ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ (Co.Re.Lab.)

Algorithm Design for Reliable Machine Learning

Διδακτορική Διατριβή

του

**Αλβέρτου Αλέξανδρου
Καλαβάση**

Αθήνα, Ιούνιος 2023

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ

Algorithm Design for Reliable Machine Learning

Διδακτορική Διατριβή

του

Αλβέρτου Αλέξανδρου
Καλαβάση

Συμβουλευτική Επιτροπή: Δημήτριος Φωτάκης (Επιβλέπων)
Χρήστος Τζάμος
Αριστείδης Παγουρτζής

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 14η Ιουνίου 2023.

....	....	....
Δημήτριος Φωτάκης Καθηγητής ΕΜΠ	Χρήστος Τζάμος Αναπληρωτής Καθηγητής ΕΚΠΑ	Αριστείδης Παγουρτζής Καθηγητής ΕΜΠ
....	....	....
Στρατής Ιωαννίδης Αναπληρωτής Καθηγητής Northeastern University	Δημήτριος Αχλιόπτας Καθηγητής ΕΚΠΑ	Μιχαήλ Λουλάκης Αναπληρωτής Καθηγητής ΕΜΠ

....
Αντώνιος Συμβώνης
Καθηγητής
ΕΜΠ

Αθήνα, Ιούνιος 2023.

....

Αλβέρτος Αλεξάνδρος Καλαβάσης
Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Abstract

In this thesis we theoretically study questions in the area of Reliable Machine Learning in order to design algorithms that are robust to bias and noise (Robust Machine Learning) and satisfy societal desiderata such as privacy and reproducibility (Responsible Machine Learning).

In the area of Robust Machine Learning, we design computationally efficient algorithms for problems in the fields of Truncated Statistics, Censored Statistics and Robust Statistics. In particular, we provide the first efficient methods for truncated distribution learning in discrete settings and perfect data sampling from truncated data. Next, we study the fundamental problem of learning from partial/coarse labels. Our main algorithmic result is that essentially any problem learnable from fine grained labels can also be learned efficiently when the coarse data are sufficiently informative. We obtain our result through a generic reduction for answering Statistical Queries (SQ) over fine grained labels given only coarse labels. We also study the central problem in Censored Statistics of Gaussian mean estimation from coarse data. Finally, we consider the problem of learning linear sorting functions in the presence of bounded noise, a problem that generalizes the problem of learning halfspaces with Massart noise.

In the area of Responsible Machine Learning, we study the notion of replicability as an algorithmic property and introduce the notion of replicable policies in the context of stochastic bandits, one of the canonical problems in interactive learning. We show that not only do replicable policies exist, but also they achieve almost the same optimal (non-replicable) regret bounds in terms of the time horizon. Lastly, we establish information-theoretic equivalences between notions of algorithmic stability such as replicability and approximate differential privacy. We do so by focusing on the following question: When two different parties use the same learning rule on their own data, how can we test whether the distributions of the two outcomes are similar? We study the similarity of outcomes of learning rules through the lens of the Total Variation (TV) distance of distributions. We say that a learning rule is TV indistinguishable if the expected TV distance between the posterior distributions of its outputs, executed on two training data sets drawn independently from the same distribution, is small. We first investigate the learnability of hypothesis classes using TV indistinguishable learners. Our main results are information-theoretic equivalences between TV indistinguishability and existing algorithmic stability notions such as replicability and approximate differential privacy.

Keywords: Theoretical Computer Science, Theoretical Machine Learning, Reliable Machine Learning

Περίληψη

Στην παρούσα διδακτορική διατριβή μελετούνται θεωρητικά προβλήματα στην περιοχή του Reliable Machine Learning με στόχο των σχεδιασμό αλγορίθμων που είναι ανθετκτικοί σε θόρυβο και μεροληψία (Robust Machine Learning) και ικανοποιούν ιδιότητες οπώς η ιδιωτικότητα και η αναπαραγωγικότητα (Responsible Machine Learning).

Στον τομέα του Robust Machine Learning, σχεδιάζουμε υπολογιστικά αποδοτικούς αλγορίθμους για προβλήματα στους τομείς των Truncated Statistics, Censored Statistics και Robust Statistics. Συγκεκριμένα, σχεδιάζουμε τις πρώτες αποδοτικές μεθόδους για μάθηση από truncated διακριτές κατανομές και παραγωγή τέλειων δειγμάτων από truncated δείγματα. Έπειτα, ασχολούμαστε με το θεμελιώδες πρόβλημα μάθησης με partial/coarse labels. Σε αυτή την κατέθυνση δίνουμε μία γενική θετική απάντηση αποδεικνύοντας πως κάθε πρόβλημα που λύνεται με Statistical Queries (Kearns 1998), μπορεί να λύθεί και με coarse labels, αν το coarsening είναι επαρκώς information preserving. Παραλληλα, απαντάμε στο ερώτημα της μάθησης του μέσου μίας Gaussian κατανομής σε υψηλές διαστάσεις από coarse δείγματα. Τέλος, μελετάμε το πρόβλημα μάθησης γραμμικών συναρτήσεων ταξινόμησης υπο την παρουσίας bounded noise, ένα πρόβλημα που γενικεύει το θεμελιώδες πρόβλημα μάθησης halfspaces με Massart noise.

Στον τομέα του Responsible Machine Learning, μελετάμε την έννοια της αναπαραγωγικότητας (replicability) ως αλγοριθμικής ιδιότητας και προτείνουμε ένα μοντέλο αναπαραγωγικότητας στον τομέα του interactive learning με εφαρμογή στο θεμελιώδες πρόβλημα των στοχαστικών bandits. Συγκεκριμένα, σχεδιάζουμε τους πρώτους replicable bandit αλγόριθμους που επιτυγχάνουν χαμηλό expected regret σε προβλήματα Multi-Armed Bandits και Linear Bandits. Παράλληλα, θεμελειώνουμε στατιστικές συνδέσεις μεταξύ της έννοιας της αναπαραγωγικότητας με αυτήν της διαφορικής ιδιωτικότητας (differential privacy). Αποδεικνύουμε πως κάθε replicable αλγόριθμος μπορεί να μετατραπεί σε ένα differentially private αλγόριθμο και ότι κάθε differentially private αλγόριθμος μπορεί να μετατραπεί σε ένα replicable αλγόριθμο.

Λέξεις Κλειδιά: Θεωρητική Πληροφορική, Θεωρία Μηχανικής Μάθησης, Αξιόπιστη Μηχανική Μάθηση

Acknowledgments

I am grateful to a bunch of amazing people for these last 3.5 years of my PhD studies. First, I would like to thank my advisors, Dimitris Fotakis and Christos Tzamos, for their endless help and unconditional support. I really enjoyed doing research with both of you and this had tremendous impact to my non-academic life.

Next, I want to thank all the members of my thesis committee: Prof. Stratis Ioannidis, Prof. Aris Pagourtzis, Prof. Dimitris Achlioptas, Prof. Michalis Loulakis and Prof. Antonis Symbonis for their help in the preparation of my thesis. I also thank Constantine Caramanis, who is the (informally speaking) the 8th member of the committee. Working with Stratis and Constantinos was a great experience and I learned a lot of things from both of you. I hope that we will keep collaborating in the future.

I am deeply grateful to two of the most amazing people and researchers that I have met in academia: Vasilis Kontonis and Manolis Zampetakis; for their patience, help and advice in various moments in these last four years. I am also thankful to the fantastic-four: Grigoris Velegkas, Vardis Kandiros, Kostas Stavropoulos and Jason Milionis; for all the great moments and memories besides what they have taught me in our (endless) meetings.

I would like to thank my awesome collaborators for making research enjoyable and fun: Dimitris Fotakis, Christos Tzamos, Konstantinos Stavropoulos, Vasilis Kontonis, Manolis Zampetakis, Jason Milionis, Stratis Ioannidis, Eleni Psaroudaki, Grigoris Velegkas, Amin Karbasi, Hossein Esfandiari, Andreas Krause, Vahab Mirrokni, Constantine Caramanis, Shay Moran, Steve Hanneke, Idan Attias.

I am glad for meeting various amazing people from the academic community that I would like to thank for great discussions, advice and stories: Andreas Galanis, Ioannis Anagnostides, Stratis Skoulakis, Manolis Vlatakis, Panagiotis Mertikopoulos, Kyriakos Lotidis, Lydia Zakynthinou, Gabriele Farina, Nikos Zarifis, Sotiris Anagnostides, Michalis Sarantis (...), Vasilis Livanos, Ilias Zadik, Tim Kunisky, Argyris Mouzakis, Gautam Kamath, Konstantina Bairaktari, Nikos Mouzakis, Giannis Fikioris, Alexia Atsidakou, Giorgos Smyrnis, Thanasis Pittas, Yuval Dagan, Fotis Iliopoulos, Vasilis Nakos, Orestis Plevrakis, Argyris Oikonomou, Giannis Panageas, Isidoros Tziotis, Konstantinos Ameranis, Evangelia Gergatsouli, Alexandros Tsigonias, Manolis Vardas, Dimitris Christou, Angeliki Giannou, Vaggos Chatziafratis, Grigoris Chrysos, Andreas Maggiori, Loukas Kavouras, Tzela Mathioudaki, Panagiotis Patsilinakos, Marianna Spurakou, Thanasis Lianeas, Natalia Kotsani, Aggela Chalki, Antonis Antonopoulos, Stathis Zachos, Greg Koumoutsos, Miltos Stouras, Spyros Dragazis, Charis Pipis, Jason Nikolaou, Radu & Christo, Davide Mazzali, Sai Ganesh Nagarajan, Alexandros Hollender, Giannis Mavrothalassitis, Panagiotis Probonas, Dionysis Kalogerias, Kostas Nikolakakis, Jane Lee, Marco Pirazzini, Xifan Yu, Anay Mehrotra, Sid Mitra, Felix Zhou, Rui Yao, Max Fishelson, Alkis Mertzios, Foivos Kalogiannis, Nikolas Patris, Christodoulos Santorinaios, Dimitris Oikonomou, Shiwei Zeng, George Dasoulas. I am sure that I forget a lot of people that deserve a place in this list and I am sorry about that. I would like to further thank all the (recent) MOP members (Ilias, Dionysis, Manos, Thodoris, Katerina, Manos) and CoReLab members for the great memories.

During my last PhD year, I was lucky enough to visit various places and universities. I want to thank Mihalis Yannakakis, Christos Papadimitriou, Rocco Servedio, Costis Daskalakis, Stratis Ioannidis, Dan Spielman, Amin Karbasi, Ola Svensson, Nicolas Flammarion and Christos Tzamos for the beautiful memories from the universities of Columbia, MIT, Northeastern, Yale, EPFL and Madison.

Finally, I am indebted to those who keep reminding me that we are creatures of a day: ΧΟΣ, ΦΟΥ, ΓΡΙ, ΤΖΙ, Teo, Stef Alif, Michalis, Verv, Miltos G., Xavi, Sotiro, Velegkas, Vardis, Kostas, Vasilis, Mouz, Konstantina, Thanasis, Labros, Zara, Savvas, Rou, Nikos, Athi, Giannis, Stou, Spuros D., John D., Jason M., Jason N., Pipis, Eleni, Tzela, Patsil, Marianna, Sonia, Despoina, Katerina R., Marina, Chris, Alex, Georgia, France, Florentia, Nikos, Ntina, Grigoris, Popi.

I also thank Lola, Lov 'N Roll, Meli and Darling for the inspiration.

Contents

1 Extended Abstract in Greek **15**

1.1 Εισαγωγή . . . 16

1.2 Robust Machine Learning . . . 16

1.3 Responsible Machine Learning . . . 17

1.4 Σύνοψη Αποτελεσμάτων . . . 17

1.4.1 Κεφάλαιο 3 - Περικομμένα δεδομένα. . . . 18

1.4.2 Κεφάλαιο 4 - Coarsened Data. . . . 18

1.4.3 Κεφάλαιο 5 - Αλλοιωμένα δεδομένα. . . . 19

1.4.4 Κεφάλαιο 6 - Replicable Bandit Algorithm Design. . . . 19

1.4.5 Κεφάλαιο 7 - Statistical Indistinguishability, Privacy and Replicability. . . . 20

1.5 Διατύπωση Βασικών Αποτελεσμάτων . . . 20

1.5.1 Βιβλιογραφικές Παρατηρήσεις . . . 22

2 Introduction **24**

2.1 Challenges in Robust Machine Learning . . . 25

2.2 Challenges in Responsible Machine Learning . . . 27

2.3 Summary of Contribution . . . 28

2.4 Technical Overview of the Thesis . . . 31

2.4.1 Learning from Truncated Samples - Chapter 3 . . . 31

2.4.2 Learning from Coarse Labels - Chapter 4 . . . 34

2.4.3 Learning with Bounded Noise - Chapter 5 . . . 39

2.4.4 Replicable Bandit Algorithms - Chapter 6 . . . 43

2.4.5 Statistically Indistinguishable Learning Algorithms - Chapter 7 45

2.5 Bibliographic Information . . . 50

3 Learning from Truncated Samples **51**

3.1 Preliminaries . . . 52

3.2 Boolean Product Distributions Truncated by Fat Sets . . . 54

3.2.1 Sampling from a Boolean Product Distribution using Samples from its Fat Truncation . . . 56

3.2.2 Parameter Estimation and Learning in Total Variation Distance . . . 57

3.2.3 Identity and Closeness Testing with Access to Truncated Samples 58
3.2.4 Learning in Total Variation Distance 59
3.2.5 Learning Ranking Distributions from Truncated Samples . . 60
3.3 Efficient Learnability from Truncated Samples: Necessary Conditions 64
3.4 PSGD for Learning Truncated Boolean Product Distributions . . . 69
3.4.1 Projected SGD: Algorithm's Description 71
3.5 Projected SGD: Theoretical Analysis 72
3.5.1 Convexity of the negative log-likelihood 72
3.5.2 Initialization Lemma 74
3.5.3 Ball in the z-space 76
3.5.4 Unbiased Estimation of the Gradient 80
3.5.5 Strong-convexity of the negative log-likelihood 81
3.5.6 Analysis of SGD 82
3.6 Appendix: Deferred Proofs 84

4 Learning from Coarse Labels 85
4.1 Classification with Coarse Data 85
4.1.1 Related Work 86
4.1.2 Technical Details for Learning from Coarse Labels 88
4.1.3 Training Models from Coarse Data 96
4.1.4 Multiclass Logistic Regression with Coarsely Labeled Data . 97
4.2 Gaussian Mean Estimation from Coarse Data 98
4.2.1 The Proofs of Theorem 4.2.2 & 4.2.3 99
4.2.2 Computational Hardness under General Partitions 99
4.2.3 Efficient Mean Estimation under Convex Partitions 109
4.2.4 Geometric Information Preservation 115

5 Learning with Bounded Noise 121
5.1 Main Definitions and Results 121
5.2 Related Work 124
5.3 Notation and Preliminaries 125
5.4 Learning in KT distance: Theorem 5.1.2 125
5.4.1 Improper Learning Algorithm 125
5.4.2 Proper Learning Algorithm: Theorem 5.1.2 127
5.5 Learning in top-r Disagreement: Theorem 5.1.3 129
5.6 Learning LSFs with Bounded Noise in Kendall's Tau distance . . . 132
5.6.1 Improperly Learning LSFs with Bounded Noise 132
5.6.2 The Proof of Theorem 5.1.2: Properly Learning LSFs with Bounded Noise 135
5.7 Learning in Top-1 Disagreement from Label Rankings 141
5.8 Learning in Top-r Disagreement from Label Rankings 147
5.8.1 The proof of Lemma 5.8.3 150
5.8.2 The proof of Lemma 5.8.4 152

5.9 Distribution-Free Lower Bounds for Top-1 Disagreement Error . . . 154
5.9.1 Top-1 Ranking Natarajan Dimension 155
5.9.2 Lower Bound for top-1 disagreement error for LSFs 156
5.10 Examples of Noisy Ranking Distributions 158

6 Replicable Bandits 160
6.1 Stochastic Bandits and Replicability 161
6.1.1 Warm-up I: Naive Replicability and Challenges 161
6.1.2 Warm-up II: Bandit Algorithms and Replicable Mean Estimation 162
6.2 Replicable Mean Estimation for Batched Bandits 163
6.3 Improved Algorithms for Replicable Stochastic Bandits 164
6.4 Replicable Stochastic Linear Bandits 166
6.4.1 Finite Action Set 166
6.4.2 Infinite Action Set 168
6.5 The Proof of Theorem 6.2.1 170
6.6 The Proof of Theorem 6.3.1 171
6.7 The Proof of Theorem 6.4.2 175
6.8 Naive Application of Algorithm 10 with Infinite Action Space . . . 177
6.9 The Proof of Theorem 6.4.6 178
6.10 Deferred Lemmata 180
6.10.1 The Proof of Lemma 6.4.4 180
6.10.2 The Proof of Lemma 6.4.5 180
6.11 Computational Performance of Algorithm 11 182

7 Statistical Indistinguishability of Learning Algorithms 186
7.1 Replicability, Differential Privacy and TV Indistinguishability . . . 186
7.2 Generalization Bounds for TV Indistinguishable Learners 187
7.3 Related Work 187
7.4 TV Indistinguishability and Replicability 191
7.5 TV Indistinguishability and Differential Privacy 195
7.6 Amplifying and Boosting TV Indistinguishable Algorithms 198
7.7 Preliminaries and Additional Definitions 200
7.7.1 Preliminaries 200
7.7.2 General Definition of Indistinguishability 202
7.7.3 Alternative Definitions of TV Indistinguishability 204
7.7.4 Coupling and Correlated Sampling 206
7.7.5 Discussion on Definition 7.1.3 207
7.8 Useful Replicable Subroutines 208
7.8.1 Replicability Preliminaries 209
7.8.2 Replicable Heavy-Hitters 210
7.8.3 Replicable Agnostic PAC Learner for Finite $\mathcal{H}$ 212
7.9 TV Indistinguishability and Replicability 213
7.9.1 The Proof of Theorem 7.4.6 213

7.9.2 A General Equivalence Result 214
7.10 TV Indistinguishability and Differential Privacy 215
7.10.1 DP Preliminaries 215
7.10.2 The Proof of Theorem 7.5.1 217
7.10.3 List-Global Stability $\implies$ TV Indistinguishability 220
7.10.4 The Proof of Theorem 7.5.2 223
7.10.5 Going Beyond Countable $\mathcal{X}$ 228
7.11 Amplification and Boosting 228
7.11.1 The Proof of Theorem 7.6.2 228
7.11.2 The Proof of Theorem 7.6.3 231
7.11.3 Tight Bound Between β, ρ 233
7.11.4 Beyond Countable $\mathcal{X}$ 234
7.12 TV Indistinguishability and Generalization 234

Chapter 1

Extended Abstract in Greek

Στην παρούσα διδακτορική διατριβή μελετούνται θεωρητικά προβλήματα στην περιοχή του Reliable Machine Learning με στόχο των σχεδιασμό αλγορίθμων που είναι ανθεκτικοί σε θόρυβο και μεροληψία (Robust Machine Learning)και ικανοποιούν ιδιότητες οπώς η ιδιωτικότητα και η αναπαραγωγικότητα (Responsible Machine Learning).

Στον τομέα του Robust Machine Learning,σχεδιάζουμε υπολογιστικά αποδοτικούς αλγορίθμους για προβλήματα στους τομείς των Truncated Statistics, Censored Statistics και Robust Statistics.Συγκεκριμένα, σχεδιάζουμε τις πρώτες αποδοτικές μεθόδους για μάθηση από truncated διακριτές κατανομές και παραγωγή τέλειων δειγμάτων από truncated δείγματα. Έπειτα, ασχολούμαστε με το θεμελιώδες πρόβλημα μάθησης με partial/coarse labels.Σε αυτή την κατέθυνση δίνουμε μία γενική θετική απάντηση αποδεικνύοντας πως κάθε πρόβλημα που λύνεται με Statistical Queries,μπορεί να λύθεί και με coarse labels,αν το coarsening είναι επαρκώς information preserving.Παραλληλα, απαντάμε στο ερώτημα της μάθησης του μέσου μίας Gaussian κατανομής σε υψηλές διαστάσεις από coarse δείγματα. Τέλος, μελετάμε το πρόβλημα μάθησης γραμμικών συναρτήσεων ταξινόμησης υπο την παρουσίας bounded noise,ένα πρόβλημα που γενικεύει το θεμελιώδες πρόβλημα μάθησης halfspaces με Massart noise. Στον τομέα του Responsible Machine Learning,μελετάμε την έννοια της αναπαραγωγικότητας (replicability) ως αλγοριθμικής ιδιότητας και προτείνουμε ένα μοντέλο αναπαραγωγικότητας στον τομέα του interactive learningμε εφαρμογή στο θεμελιώδες πρόβλημα των στοχαστικών bandits. Συγκεκριμένα, σχεδιάζουμε τους πρώτους replicable bandit αλγόριθμους που επιτυγχάνουν χαμηλό expected regret σε προβλήματα Multi-Armed Bandits και Linear Bandits.Παράλληλα, θεμελειώνουμε στατιστικές συνδέσεις μεταξύ της έννοιας της αναπαραγωγικότητας με αυτήν της διαφορικής ιδιωτικότητας differential privacy. Αποδεικνύουμε πως κάθε replicable αλγόριθμος μπορεί να μετατραπεί σε ένα differentially private αλγόριθμο και ότι κάθε differentially private αλγόριθμος μπορεί να μετατραπεί σε ένα replicable αλγόριθμο.

1.1 Εισαγωγή

Η Μηχανική Μάθηση αποτελεί θεμελιώδες δομικό στοιχείο της σύγχρονης ζωής με πολυάριθμες εφαρμογές. Ο αντίκτυπός της είναι εκτεταμένος και παρατηρείται συχνά σε τομείς όπως η υγειονομική περίθαλψη, οι μεταφορές, τα οικονομικά και η εκπαίδευση. Για παράδειγμα, η Μηχανική Μάθηση μπορεί να χρησιμοποιηθεί για τη βελτίωση της υγειονομικής περίθαλψης μέσω καλύτερων διαγνώσεων και εξατομικευμένης θεραπείας και για τη μείωση της κυκλοφοριακής συμφόρησης χρησιμοποιώντας προγνωστικά αναλυτικά στοιχεία. Ενώ τέτοιες εφαρμογές της Επιστήμης Υπολογιστών και Δεδομένων μπορούν να οδηγήσουν σε μια ασφαλέστερη, πιο αποτελεσματική και πιο δίκαιη κοινωνία, υπάρχουν διάφοροι κίνδυνοι που κρύβονται πίσω από μια τέτοια τεχνολογία. Διάφορες πηγές προκατάληψης ή κακόβουλων επιθέσεων μπορούν να επηρεάσουν την ποιότητα, την απόδοση, τη δικαιοσύνη και το απόρρητο των συστημάτων Μηχανικής Μάθησης. Ως εκ τούτου, είναι πρωταρχικής σημασίας η παροχή αξιόπιστων μοντέλων μηχανικής εκμάθησης που διασφαλίζουν την καλή συμπεριφορά των αναπτυγμένων συστημάτων με αποδεδειγμένες εγγυήσεις όσον αφορά την ευρωστία και την απόδοση.

Η παρούσα διατριβή εστιάζει στο σχεδιασμό και την απόκτηση τυπικών θεωρητικών εγγυήσεων σχετικά με τέτοια αξιόπιστα συστήματα που διασφαλίζουν (ι) ανθεκτικότητα στην προκατάληψη, (ιι) διαφορική ιδιωτικότητα και (ιιι) αναπαραγωγισιμότητα.

1.2 Robust Machine Learning

Πολλές κοινώς χρησιμοποιούμενες στατιστικές μέθοδοι βασίζονται σε μια κρίσιμη υπόθεση: ότι τα δεδομένα κατανέμονται ανεξάρτητα και πανομοιότυπα (i.i.d.).Σύμφωνα με αυτή την υπόθεση, κάθε δείγμα λαμβάνεται υπό συνεπείς συνθήκες και δεν επηρεάζει τα υπόλοιπα δείγματα. Ωστόσο, αυτή η υπόθεση δεν λαμβάνει υπόψη τις διάφορες προκλήσεις στη διαδικασία συλλογής δεδομένων, οδηγώντας σε μεροληπτικά σύνολα δεδομένων. Η παρουσία τέτοιας μεροληψίας μπορεί να αποφέρει παραπλανητικά ή άδικα στατιστικά συμπεράσματα. Κατά συνέπεια, καθίσταται σημαντικό να εντοπιστούν οι πηγές της μεροληψίας και, το πιο σημαντικό, να επινοηθούν στρατηγικές για τη διεξαγωγή στατιστικών αναλύσεων παρουσία μεροληψίας. Αυτό το ζήτημα είναι ένα θεμελιώδες πρόβλημα με εκτεταμένες εφαρμογές σε διάφορα επιστημονικά πεδία, συμπεριλαμβανομένης της Ιατρικής Επιστήμης και της Οικονομίας. Ο τομέας του Robust Machine Learning στοχεύει να ασχοληθεί με τέτοια προκατειλημμένα σύνολα δεδομένων και αλγόριθμους σχεδιασμού που αποδεδειγμένα αντιμετωπίζουν τέτοια φαινόμενα.

Στο πρώτο μέρος αυτής της διατριβής, παρέχουμε θεωρητική κατανόηση όταν τα δεδομένα είναι προκατειλημμένα λόγω των ακόλουθων σημαντικών και αναδυόμενων προκλήσεων: (1) *truncation*, (2) *coarsening* και (3) *corruptions με ημιτυχαίο θόρυβο.*

1.3 Responsible Machine Learning

Καθώς η Μηχανική Μάθηση γίνεται όλο και περισσότερο μέρος των εφαρμογών της πραγματικής ζωής, οι επιστήμονες δεδομένων στοχεύουν στην ανάπτυξη μοντέλων και αλγορίθμων Μηχανικής Μάθησης με τρόπο που να ευθυγραμμίζεται με τις ηθικές αρχές και αξίες και με τρόπο ώστε τα παρεχόμενα αποτελέσματα να είναι αξιόπιστα και έγκυρα.

Το ζήτημα της ιδιωτικότητας κατά την ανάλυση δεδομένων έχει πλούσιο ιστορικό υπόβαθρο που περιλαμβάνει διάφορους τομείς σπουδών. Με την προηγμένη τεχνολογία που επιτρέπει την όλο και πιο ισχυρή συλλογή και οργάνωση δεδομένων χρηστών, υπάρχει μια αυξανόμενη ζήτηση για μια αυστηρή και καλά καθορισμένη έννοια της ιδιωτικής ζωής. Η έννοια της *διαφορικής ιδιωτικότητας* αναδύεται ως θεμελιώδης λύση στους κλάδους, παρέχοντας έναν τρόπο προστασίας του απορρήτου των δεδομένων ακόμα και όταν συλλέγονται δεδομένα από μια μεγάλη ομάδα ατόμων. Διασφαλίζει ότι τα άτομα δεν μπορούν να εντοπιστούν μέσω των δεδομένων τους, ενώ ταυτόχρονα παρέχει πολύτιμες πληροφορίες που μπορούν να βελτιώσουν τις υπηρεσίες.

Η αναπαραγωγισιμότητα είναι σημαντική γιατί επιτρέπει σε άλλους να επαληθεύουν και να επικυρώνουν τα ευρήματα οποιασδήποτε έρευνας ή πειράματος. Χωρίς αναπαραγωγισιμότητα, τα ευρήματα της έρευνας δεν είναι αξιόπιστα και η επιστημονική πρόοδος καταπνίγεται. Συγκεκριμένα, προκειμένου τα επιστημονικά ευρήματα να είναι έγκυρα και αξιόπιστα, η πειραματική διαδικασία πρέπει να είναι επαναλήψιμη και πρέπει να παρέχει συνεκτικά αποτελέσματα και συμπεράσματα σε αυτές τις επαναλήψεις. Εν ολίγοις, η αναπαραγωγισιμότητα μπορεί να μειώσει την προκατάληψη και να αυξήσει τη διαφάνεια στην έρευνα, γεγονός που μπορεί να βοηθήσει να διασφαλιστεί ότι τα αποτελέσματα είναι δίκαια και ακριβή. Στην πραγματικότητα, η έλλειψη αναπαραγωγισιμότητας ήταν ένα σημαντικό ζήτημα σε πολλούς επιστημονικούς τομείς, που συνήθως αναφέρονται ως την «κρίση αναπαραγωγισιμότητας». Μια έρευνα του 2016 που εμφανίστηκε στο Nature (Bak16b)αποκάλυψε ότι περισσότερο από το 70% των ερευνητών απέτυχαν στην προσπάθειά τους να αναπαράγουν τα πειράματα ενός άλλου ερευνητή.

Στο δεύτερο μέρος αυτής της διατριβής, εμείς κάνουμε συνεισφορές στον τομέα της Υπεύθυνης Μηχανικής Μάθησης: καθιερώνουμε αυστηρές συνδέσεις μεταξύ των προαναφερθέντων εννοιών της (1) *διαφορικής ιδιωτικότητας* και της (2)*αναπαραγωγισιμότητας* και σχεδιάζουμε αλγόριθμους που επιλύουν σημαντικά στατιστικά προβλήματα υπό περιορισμούς αναπαραγωγισιμότητας.

1.4 Σύνοψη Αποτελεσμάτων

Σε αυτή την ενότητα συνοψίζουμε τα αποτελέσματα αυτής της διατριβής και παρουσιάζουμε μια επισκόπηση της δομής της διατριβής. Ξεκινάμε εξηγώντας σύντομα το περιεχόμενο καθενός από τα κεφάλαια. Το πρώτο μισό της διατριβής ασχολείται με τη Στιβαρή Μηχανική Μάθηση, ενώ το δεύτερο χειρίζεται προβλήματα που σχετίζονται με την Υπεύθυνη Μηχανική Μάθηση.

Στο πρώτο μέρος της διατριβής (Chapter 3-5),θα ασχοληθούμε με ερωτήσεις σχετικά με τη Στιβαρή Μηχανική Μάθηση. Ειδικότερα, τα τρία πρώτα κεφάλαια οργανώνονται ως εξής.

1.4.1 Κεφάλαιο 3 - Περικομμένα δεδομένα.

Αρχικά μελετάμε το πρόβλημα της εκτίμησης των παραμέτρων μιας κατανομής γινομένου Boole σε d διαστάσεις, όταν τα δείγματα περικόπτονται από ένα σύνολο $S \subseteq \{0,1\}^d$ προσβάσιμο μέσω ενός membership oracle.Εισάγουμε μια φυσική έννοια του πάχους του συνόλου περικοπής S, σύμφωνα με την οποία τα περικομμένα δείγματα αποκαλύπτουν αρκετές πληροφορίες σχετικά με την πραγματική κατανομή. Δείχνουμε ότι εάν το σύνολο περικοπής είναι επαρκώς παχύ, τα δείγματα από την πραγματική κατανομή μπορούν να δημιουργηθούν από περικομμένα δείγματα. Μια εκπληκτική συνέπεια είναι ότι σχεδόν οποιοδήποτε στατιστική εργασία που μπορεί να εκτελεστεί αποτελεσματικά για κατανομές γινόμενα Boole, μπορούν επίσης να πραγματοποιηθούν από περικομμένα δείγματα, με μικρή αύξηση σε πολυπλοκότητα δείγματος. Εξερευνώντας τα όρια εκμάθησης διακριτών μοντέλων από περικομμένα δείγματα, προσδιορίζουμε τις φυσικές συνθήκες που είναι απαραίτητες για αποτελεσματική ταυτοποίηση. Προσαρμόζοντας προσεκτικά την προσέγγιση Στοχαστικής Κάθοδος Κλίσης του (DGTZ18),δείχνουμε ότι αυτές οι συνθήκες είναι επίσης επαρκείς για την αποτελεσματική εκμάθηση των περικομμένων κατανομών.

1.4.2 Κεφάλαιο 4 - Coarsened Data.

Σε αυτό το κεφάλαιο, ορίζουμε και μελετάμε επίσημα το πρόβλημα της μάθησης από μερικά δεδομένα (Challenge 2).Αντί να παρατηρούμε τις πραγματικές ετικέτες από ένα σύνολο $\mathcal{Z}$, παρατηρούμε χονδροειδείς ετικέτες που αντιστοιχούν σε μια κατάτμηση $\mathcal{Z}$ (ή σε ένα μείγμα κατατμήσεων). Το κύριο αλγοριθμικό μας αποτέλεσμα είναι ότι ουσιαστικά οποιοδήποτε πρόβλημα μαθαίνεται από fine ετικέτες, μπορεί επίσης να μαθευτεί αποτελεσματικά και όταν τα coarse δεδομένα είναι επαρκώς ενημερωτικά. Λαμβάνουμε το αποτέλεσμά μας μέσω μιας γενικής αναγωγής για την απάντηση στα στατιστικά ερωτήματα (SQ) έναντι των fine ετικετών που δίνονται μόνο σε coarse ετικέτες. Ο αριθμός των coarse ετικετών που απαιτούνται εξαρτάται πολυωνυμικά από την παραμόρφωση των πληροφοριών που οφείλεται στη χονδροποίηση και τον αριθμό των λεπτών ετικετών $|\mathcal{Z}|$. Επίσης, ερευνούμε ένα κεντρικό πρόβλημα σε λογοκριμένα στατιστικά: αυτό της εκτίμησης του Gaussian μέσου όρου από coarse δεδομένα. Παρέχουμε έναν αποτελεσματικό αλγόριθμο όταν τα σύνολα είναι κυρτά και δείχνουμε ότι το πρόβλημα είναι NP-hard ακόμη και για πολύ απλά μη κυρτά σύνολα. Από τεχνικής πλευράς, το αλγοριθμικό μας αποτέλεσμα βασίζεται στην κομψή ανισότητα Brascamp–Lieb και το hardness αποτέλεσμα μας βασίζεται σε (ίσως απροσδόκητες) συνδέσεις μεταξύ coarse κατανομών Gauss και του θεμελιώδους προβλήματος της εύρεσης της μέγιστης τομής σε ένα γράφημα, το οποίο είναι γνωστό ότι είναι NP-hard ακόμα και κατά προσέγγιση.

1.4.3 Κεφάλαιο 5 - Αλλοιωμένα δεδομένα.

Η κατάταξη ετικετών είναι η εποπτευόμενη εργασία εκμάθησης μιας συνάρτησης ταξινόμησης που αντιστοιχίζει τα χαρακτηριστικά διανύσματα $\boldsymbol{x} \in \mathbb{R}^d$ στις ταξινομήσεις $\sigma(\boldsymbol{x}) \in \mathbb{S}_k$ σε ένα πεπερασμένο σύνολο από k ετικέτες. Εστιάζουμε στη θεμελιώδη περίπτωση της εκμάθησης συναρτήσεων γραμμικής ταξινόμησης (LSF) κάτω από τα περιθώρια Gauss:$\boldsymbol{x}$ λαμβάνεται ως δείγμα από την d-dimensional standard normal και η βασική κατάταξη αλήθειας $\sigma^\star(\boldsymbol{x})$ είναι η σειρά που προκαλείται από την ταξινόμηση των συντεταγμένων του διανύσματος $\boldsymbol{W}^\star \boldsymbol{x}$, όπου $\boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$. Θεωρούμε την εκμάθηση LSF παρουσία περιορισμένου θορύβου (Challenge 3): υποθέτοντας ότι ένα αθόρυβο παράδειγμα είναι της μορφής $(\boldsymbol{x}, \sigma^\star(\boldsymbol{x}))$, παρατηρούμε $(\boldsymbol{x}, \pi)$, όπου για οποιοδήποτε ζεύγος στοιχείων $i \neq j$, η πιθανότητα ότι η σειρά των i, j είναι διαφορετική στο π από ότι στο $\sigma^\star(\boldsymbol{x})$ είναι *το πολύ* $\eta < 1/2$. Σχεδιάζουμε αποδοτικούς αλγόριθμους μάθησης που μαθαίνουν υποθέσεις εντός κανονικοποιημένης απόστασης Kendall's Tau ϵ από τη βασική αλήθεια με $N = \widetilde{O}(d \log(k)/\epsilon)$ παραδείγματα και χρόνο εκτέλεσης $\mathrm{poly}(N, k)$. Για την πιο απαιτητική loss function top-r, δίνουμε έναν αποδοτικό αλγόριθμο μάθησης που επιτυγχάνει ϵ τοπ-r διαφωνία με τη βασική αλήθεια με $N = \widetilde{O}(dkr/\epsilon)$ δείγματα και $\mathrm{poly}(N)$ χρόνο εκτέλεσης.

Στο δεύτερο μέρος της διατριβής (Chapter 6-7),μελετάμε προβλήματα που αφορούν την Υπεύθυνη Μηχανική Μάθηση. Τα κεφάλαια οργανώνονται ως εξής.

1.4.4 Κεφάλαιο 6 - Replicable Bandit Algorithm Design.

Εισάγουμε την έννοια των αναπαραγόμενων πολιτικών στο πλαίσιο των στοχαστικών bandits , ένα από τα κανονικά προβλήματα στη διαδραστική μάθηση. Μια πολιτική στο περιβάλλον bandits ονομάζεται αναπαραγώγιμη εάν τραβάει, με μεγάλη πιθανότητα, την *ακριβώς* ίδια ακολουθία arms σε δύο διαφορετικές και ανεξάρτητες εκτελέσεις (δηλαδή υπό ανεξάρτητες πραγματοποιήσεις ανταμοιβής). Δείχνουμε ότι όχι μόνο υπάρχουν αναπαραγόμενες πολιτικές, αλλά και επιτυγχάνουν σχεδόν τα ίδια βέλτιστα (μη αναπαραγόμενα) όρια regret όσον αφορά το χρονικό ορίζοντα. Πιο συγκεκριμένα, στο πλαίσιο των στοχαστικών bandits πολλαπλών arms , αναπτύσσουμε μια πολιτική με βέλτιστο regret που εξαρτάται από το πρόβλημα του οποίου η εξάρτηση από την παράμετρο αναπαραγωγιμότητας είναι επίσης βέλτιστη. Ομοίως, για στοχαστικούς γραμμικούς bandits (με πεπερασμένους και άπειρους arms) αναπτύσσουμε αναπαραγόμενες πολιτικές που επιτυγχάνουν τα πιο γνωστά όρια regret ανεξάρτητα από το πρόβλημα με βέλτιστη εξάρτηση από την παράμετρο αναπαραγωγιμότητας. Τα αποτελέσματά μας δείχνουν ότι, παρόλο που η τυχαιοποίηση είναι ζωτικής σημασίας για την αντιστάθμιση εξερεύνησης-εκμετάλλευσης, μπορεί να επιτευχθεί μια βέλτιστη ισορροπία ενώ τραβάμε τα ίδια ακριβώς arms σε δύο διαφορετικούς γύρους εκτελέσεων.

Summary of Results			
Setting	Algorithm	Regret	Theorem
Stochastic MAB	Algorithm 8	$\widetilde{O}\left(\frac{K^2\log^3(T)H_\Delta}{\rho^2}\right)$	Theorem 6.2.1
Stochastic MAB	Algorithm 9	$\widetilde{O}\left(\frac{K^2\log(T)H_\Delta}{\rho^2}\right)$	Theorem 6.3.1
Stochastic Linear Bandits	Algorithm 10	$\widetilde{O}\left(\frac{K^2\sqrt{dT}}{\rho^2}\right)$	Theorem 6.4.2
Stochastic Linear Bandits Infinite Action Space	Algorithm 11	$\widetilde{O}\left(\frac{\text{poly}(d)\sqrt{T}}{\rho^2}\right)$	Theorem 6.4.6

1.4.5 Κεφάλαιο 7 - Statistical Indistinguishability, Privacy and Replicability.

Όταν δύο διαφορετικές εκτελέσεις χρησιμοποιούν τον ίδιο κανόνα μάθησης στα δικά τους δεδομένα, πώς μπορούμε να ελέγξουμε εάν οι κατανομές των δύο αποτελεσμάτων είναι παρόμοιες· Σε αυτό το κεφάλαιο, μελετάμε την ομοιότητα των αποτελεσμάτων των κανόνων μάθησης μέσα από την Total Variationαπόσταση κατανομών. Λέμε ότι ένας κανόνας μάθησης δεν διακρίνεται στην TV απόστασταση εάν η αναμενόμενη TV απόσταση μεταξύ των posterior κατανομών των εξόδων του, που εκτελούνται σε δύο σύνολα δεδομένων εκπαίδευσης που έχουν σχεδιαστεί ανεξάρτητα από την ίδια κατανομή, είναι μικρή. Αρχικά διερευνούμε τη δυνατότητα εκμάθησης χρησιμοποιώντας TV αδιάκριτους αλγόριθμους. Τα κύρια αποτελέσματά μας είναι οι θεωρητικές-στατιστικές ισοδυναμίες μεταξύ της TV δυσδιάκρισης και των υπαρχουσών εννοιών αλγοριθμικής σταθερότητας, όπως η δυνατότητα αναπαραγωγής και η διαφορική ιδιωτικότητα.

1.5 Διατύπωση Βασικών Αποτελεσμάτων

Σε αυτή την ενότητα παρουσιάζεται μία λίστα με τα βασικά αποτελέσματα της διδακτορικής διατριβής.

Informal Theorem. *Με ένα εκτιμόμενο αριθμό $O(\log(d)/\alpha)$ δειγμάτων από μία α-fat truncation μίας κατανομής γινομένου Boole $\mathcal{D}$, μπορούμε να παράξουμε ένα δείγμα $\boldsymbol{x} \in \{0,1\}^d$ κατανεμημένο όπως η $\mathcal{D}$.*

Informal Theorem. *Κάτω από τις υποθέσεις (1) - (4), υπάρχει αλγόριθμος που υπολογίζει μία εκτίμηση $\widehat{\boldsymbol{z}}$ του logit διανύσματος $\boldsymbol{z}$ της αληθινής κατανομής $\mathcal{D}$*

τέτοιο ώστε $\|\boldsymbol{z} - \widehat{\boldsymbol{z}}\|_2 \leq \epsilon$ *με πιθανότητα τουλάχιστον* $1 - \delta$, *και επιτυγχάνει χρονική και δειγματική ποπυπλοκότητα πολυωνυμική στο* d, $1/\epsilon$ *και* $\log(1/\delta)$.

Informal Theorem. *Κάθε κλάση* $\mathcal{C} \subseteq [k]^{\mathcal{X}}$ *που μαθαίνεται αποδοτικά από* M *statistical queries από finely labeled examples* $(x, z) \sim \mathcal{D}$, *μαθαίνεται αποδοτικά και από* $O(\mathrm{poly}(k/\alpha)) \cdot M$ *coarsely labeled examples* $(x, S) \sim \mathcal{D}_\pi$ *κάτω από κάθε* α*-information preserving partition distribution* π.

Informal Theorem. *Έστω* π *μία γενική κατανομή διαμέρισης. Εκτός εάν* $\mathrm{RP} = \mathrm{N\Pi}$, *κανένας αλγόριθμος με πρόσβαση στην* $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, *δεν μπορεί σε χρόνο* $\mathrm{poly}(d)$, *να υπολογίσει ενα* $\widetilde{\boldsymbol{\mu}} \in \mathbb{R}^d$ *τέτοιο ώστε* $d_{\mathrm{TV}}(\mathcal{N}_\pi(\widetilde{\boldsymbol{\mu}}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star)) < 1/d^c$ *για μία απόλυτη σταθερά* $c > 1$.

Informal Theorem. *Έστω* $\epsilon \in (0, 1)$ *και η κατανομή* $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$ *σε* d *διαστάσεις. Υποθέτουμε ότι η κατανομή διαμέρισης* π *είναι* α*-information preserving και είναι supported σε κυρτές διαμερίσεις του* $\mathbb{R}^d$. *Τότε υπάρχει αλγόριθμος που χρησιμοποιεί* $N = \widetilde{O}(d/(\epsilon^2\alpha^2))$ *δείγματα από την* $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$ *και υπολογίζει μία εκτίμηση* $\widetilde{\boldsymbol{\mu}}$ *που ικανοποιεί* $d_{\mathrm{TV}}(\mathcal{N}(\widetilde{\boldsymbol{\mu}}), \mathcal{N}(\boldsymbol{\mu}^\star)) \leq \epsilon$, *με πιθανότητα τουλάχιστον* 99%.

Informal Theorem. *Έστω* $\eta \in [0, 1/2)$ *και* $\epsilon, \delta \in (0, 1)$. *Έστω* $\mathcal{D}$ *μία* η*-noisy linear label ranking κατανομή που ικανοποιεί τον ορισμό 5.1.1 με αληθινό LSF* $\sigma_{\boldsymbol{W}^\star}(\cdot)$. *Τότε υπάρχει αλγόριθμος που χρησιμοποιεί* $N = \widetilde{O}\left(\frac{d}{\epsilon(1-2\eta)^6} \log(k/\delta)\right)$ *δείγματα από την* $\mathcal{D}$, *και σε χρόνο πολυωνιμικό στο πλήθος των δειγμάτων υπολογίζει έναν πίνακα* $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ *ώστε με πιθανότητα τουλάχιστον* $1 - \delta$,

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d}[\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq \epsilon\,.$$

Informal Theorem. *Έστω* $\rho \in (0, 1), T \in \mathbb{N}$ *και* $H_\Delta = \sum_{j:\Delta_j > 0} 1/\Delta_j$, *όπου* Δ_j *είναι η διαφορά μεταξύ της επιλογής* j *και της βέλτιστης.*

1. *Υπάρχει ένας* ρ*-αναπαραγωγίσιμος αλγόριθμος για το stochastic MAB setting με* K *arms με αναμενόμενο regret*

$$\widetilde{O}(K^2 \log(T) H_\Delta/\rho^2)\,.$$

2. *Υπάρχει ένας ρ-αναπαραγωγίσιμος αλγόριθμος για το stochastic d-dimensional linear bandit setting με K arms με αναμενόμενο regret*

$$\widetilde{O}(K^2\sqrt{dT}/\rho^2)\,.$$

Informal Theorem. *Τα παρακάτω είναι αληθή.*

- *Εάν ένας κανόνας μάθησης A είναι n-sample ρ-replicable, τότε είναι και n-sample ρ-TV indistinguishable.*
- *Έστω $\mathcal{X}$ μετρήσιμο και A ένας κανόνας μάθησης που είναι n-sample ρ-TV indistinguishable. Τότε υπάρχει ισοδύναμος κανόνας A' που είναι n-sample $\frac{2\rho}{1+\rho}$-replicable.*

Informal Theorem. *Τα παρακάτω είναι αληθή.*

- *Εστω $\gamma \in (0, 1/2), \alpha, \beta, \rho \in (0, 1)^3$. Αν η $\mathcal{H}$ είναι learnable by an n-sample $(1/2-\gamma, 1/2-\gamma)$-accurate $(0.1, 1/(n^2 \log(n)))$-differentially private learner, τότε είναι learnable by an (α, β)-accurate ρ-TV indistinguishable learning rule.*
- *Έστω $\mathcal{X}$ μετρήσιμο. Αν η $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ είναι learnable by an (α, β)-accurate ρ-TV indistinguishable learner A, για κάποιο $\rho \in (0,1), \alpha \in (0, 1/2), \beta \in \left(0, \frac{1-\rho}{1+\rho}\right)$, τότε για κάθε $(\alpha', \beta', \varepsilon, \delta) \in (0,1)^4$, είναι learnable by an $(\alpha+\alpha', \beta')$-accurate (ε, δ)-differentially private learner A'.*

1.5.1 Βιβλιογραφικές Παρατηρήσεις

Τα αποτελέσματα που περιγράφονται σε αυτή τη διατριβή έχουν ήδη εμφανιστεί σε υπάρχουσες δημοσιεύσεις, τις οποίες εμείς αναφέρω εν συντομία παρακάτω.

Το κεφάλαιο 3 βασίζεται στο (FKT20) που παρουσιάστηκε στο COLT 2020. Το κεφάλαιο 4 βασίζεται στο (FKKT21) που παρουσιάστηκε στο COLT 2021. Το κεφάλαιο 5 βασίζεται στο (FKKT22) που παρουσιάστηκε στο NeurIPS 2022. Το κεφάλαιο 6 βασίζεται στο (EKM$^+$23) που παρουσιάστηκε στο ICLR 2023. Το κεφάλαιο 7 βασίζεται στο (KKMV23) που παρουσιάστηκε στο ICML 2023.

Άλλες εργασίες του συγγραφέα κατά τη διάρκεια του διδακτορικού του που δεν περιλαμβάνονται στην παρούσα διατριβή είναι (FKS21; FKP21; KSZ22; MKFI22; FKT22; KVK22).

Chapter 2

Introduction

Machine Learning constitutes a fundamental building block of modern life with numerous applications. Its impact is far-reaching and often seen in areas such as healthcare, transportation, finance, and education. For instance, Machine Learning can be used to improve healthcare through better diagnoses and personalized treatment and to reduce traffic congestion using predictive analytics. While such applications of Computer and Data Science can lead to a safer, more efficient, and more equitable society, there exist various perils hidden behind such technology. Various sources of bias or malicious attacks can influence the quality, performance, fairness and privacy of Machine Learning systems. Hence it is of primal importance to provide Reliable Machine Learning models assuring the well-behavior of the deployed systems with provable guarantees in terms of robustness and performance.

This thesis focuses on designing and obtaining formal theoretical guarantees about such reliable systems that ensure (i) robustness to bias, (ii) differential privacy and (iii) reproducibility.

Robust Machine Learning. Many commonly used statistical methods rely on a crucial assumption: that data points are independently and identically distributed (i.i.d.). According to this assumption, each sample is drawn under consistent conditions and does not impact the rest of the samples. However, this assumption fails to account for various challenges in the data collection process, leading to biased datasets. The presence of such bias can yield misleading or unfair statistical conclusions. Consequently, it becomes essential to identify the sources of bias and, more importantly, devise strategies for conducting statistical analyses in the presence of bias. This issue is a fundamental problem with widespread applications in diverse scientific fields, including Medical Science and Economics. The area of Robust Machine Learning aims to deal with such biased datasets and design algorithms that provably tackle such phenomena.

In the first part of this thesis, we provide theoretical understanding when the data are biased due the following significant and emerging challenges: (1) *truncation*, (2) *coarsening* and (3) *corruptions with semi-random noise.*

Responsible Machine Learning. As Machine Learning increasingly becomes part of real-life applications, data scientists aim at developing and deploying Machine Learning models and algorithms in a manner that aligns with ethical principles and values and in a way that the provided results are reliable and valid.

The issue of privacy during data analysis has a rich historical background that encompasses various fields of study. With advancing technology enabling increasingly potent collection and organization of user data, there is a growing demand for a rigorous and well-defined concept of privacy. The notion of *differential privacy* emerges as a fundamental solution in industries, providing a way to protect data privacy even when collecting data from a large group of individuals. It ensures that individuals cannot be identified through their data, while simultaneously providing valuable insights that can enhance services.

Reproducibility is important because it allows others to verify and validate the findings of any research or experiment. Without reproducibility, research findings cannot be trusted and scientific progress is stifled. In particular, in order for scientific findings to be valid and reliable, the experimental process must be repeatable, and must provide coherent results and conclusions across these repetitions. In short, reproducibility can reduce bias and increase transparency in research, which can help to ensure that results are fair and accurate. In fact, lack of reproducibility has been a major issue in many scientific areas, commonly referred to as the “reproducibility crisis”; a 2016 survey that appeared in Nature (Bak16b) revealed that more than 70% of researchers failed in their attempt to reproduce another researcher’s experiments.

In the second part of this thesis, we make several contributions in the area of Responible Machine Learning: we establish rigorous connections between the aforementioned notions of (1) *differential privacy* and (2)*reproducibility*, and, design algorithms that solve important statistical problems under reproducibility constraints.

2.1 Challenges in Robust Machine Learning

In the first part of the thesis, we opt to design computationally efficient algorithms that are robust to realistic noise models and, in particular, truncation, coarsening and semi-random corruptions.

Truncated Data. Inference from truncated or censored samples is a classical challenge in Statistics. Truncation occurs when samples falling outside of some subset S of the support of the distribution are not observed, and their count in proportion to the observed samples is also not observed. Censoring is similar but simpler; the fraction of samples falling outside of S is given to the data analyst. Truncation and censoring of samples have unlimited manifestations in economics, engineering, quality control, medical and biological sciences, social sciences, and all areas of the physical sciences. See (DGTZ18; DGTZ19a) for various historical references and Gil Kalai's blog for an interesting anecdote on truncated statistics and Henri Poincaré.

Let us provide an intuitive example. Consider the statistical task where Alice wants to estimate the mean height of a group of people. Assume that the data are collected by Bob, a malicious adversary who wants to cause trouble to Alice; Bob erases any measurement that is smaller than 190cm and provides the "corrupted" dataset to Alice. Hence, Alice observes a truncated version of the true dataset and, under the natural assumption that the heights of the group are normally distributed with mean at 180cm, a naive empirical estimate will be completely far from the true mean. Alice can understand that the dataset is biased since she is not observing the expected "bell curve" and so she is suspicious that Bob has performed a malicious action. Can Alice use only this truncated dataset to obtain the correct estimation, i.e., output a good approximation of the true mean? The area of truncated statistics aims to answer exactly that kind of questions. In short, the answer is "yes". Intuitively, Alice can "reconstruct" the Gaussian curve by simply observing only a part (say the tail) of the distribution's support. Is this possible to be done efficiently in high-dimensions? This question falls in the realm of computationally efficient inference from truncated samples and has been raised and studied in various works (DGTZ18; KTZ19; DGTZ19a; NP19).

Challenge 1. *Design computationally efficient algorithms that are robust in the presence of truncated data in high dimensions.*

Coarsened Data. Another classical challenge in Statistics is the problem of estimation from coarsened data (Tsi06). Recall the game between Alice and Bob. Now Bob instead of hiding samples below 190cm performs the following strategy: He observes each drawn sample x and replaces it with the closest to x multiple of 50. Hence, the sample $x = 151$cm becomes 150cm and the sample $x = 176$cm becomes 200cm. The terminology for Bob's rounding strategy in statistics comes by the name of *coarsening*. Alice observes only the coarsened datapoints. Bob's strategy is perhaps the most elementary way to coarsen data; he simply rounds each point to a desired accuracy. We remark that there is an equivalently conceptual way to think of coarsening. Alice does not observe the actual points x but sets S (in this case intervals) that contain the true observation, e.g., Alice's observation 200cm is the interval $[175, 225)$cm, which contains the actual observation 176cm. Can Alice retrieve the correct mean? Once again, is this possible to be done in a computationally efficient way in high-dimensions? Designing algorithms from coarsened data constitute a classical challenge in Computer Science (GLB$^+$18; CDCM18; TSD$^+$20; QCJ$^+$20; LGW17; JLYW19) and Statistics (CST11a; CSGGSR14; FLH$^+$20; CRB20; LXF$^+$20; WCH$^+$21).

Challenge 2. *Design computationally efficient algorithms that are robust in the presence of coarsened data in high-dimensions.*

Corrupted Data. The field of Robust Statistics focuses on addressing the general challenge of developing estimators that maintain good performance even when the data significantly deviates from idealized modeling assumptions (DK19).

The study of robust statistical procedures can be traced back to the seminal works of (Tuk60) and (Hub64). Classical statistical theory has provided insights into the information-theoretic limits of robust estimation for many common problems. However, the computational aspects of this field remained poorly understood until relatively recently. In particular, the first computational results in robust statistics appeared in (DKK$^+$19b) and (LRV16a). After these breakthrough results, numerous computationally efficient robust estimators for high-dimensional learning tasks have been designed. It is worth noting that there exists a wide range of models available that capture various forms of data corruptions, contributing to the versatility of robust statistical techniques. In this thesis, we will focus on the semi-random model that interpolates between the fully-random model and the fully-adversarial model (see e.g., (MN06) for details). The main motivation behind semi-random noise models is that they are expressive enough to capture various real-world scenarios (compared to the fully-random models; see e.g., (DGT19a)) and potentially allow for computationally efficient algorithms (compared to the fully-adversarial models; see e.g., (FGRW12)).

Challenge 3. *Design computationally efficient algorithms that are robust in the presence of (semi-randomly) corrupted data in high-dimensions.*

The above challenge has motivated an extensive line of work (MN06; Vap06; Slo88; Slo92; RS94; Slo96; DGT19b; CKMY20; DKT21; ABHU15; ABHZ16; YZ17; ZLC17; BZ17; MV19; DKTZ20; ZSA20; ZL21).

2.2 Challenges in Responsible Machine Learning

This second part of the thesis lies in the fundamental research direction of Responsible Machine Learning. In this thesis, we will focus on two notions of emerging importance in Data Science: (1) replicability and (2) differential privacy.

Lack of replicability in experiments has been a major issue, usually referred to as the *reproducibility crisis*, in many scientific areas such as biology and chemistry. As we have already mentioned, the results of a survey that appeared in Nature (Bak16a) are very worrisome: more than 70% of the researchers that participated in it could not replicate other researchers' experimental findings while over half of them were not able to even replicate their own conclusions. In the past few years the number of scientific publications in the Machine Learning (ML) community has increased exponentially. Significant concerns and questions regarding replicability have also recently been raised in the area of ML. This can be witnessed by the establishment of various reproducibility challenges in major ML conferences such as the ICLR 2019 Reproducibility Challenge (PSF$^+$19) and the NeurIPS 2019 Reproducibility Program (PVLS$^+$21).

Motivated by this crucial problem, the theoretical Machine Learning community initiated the study of reproducibility/replicability as a property of learning algorithms. Inspired by the notion of pseudo-deterministi algorithms in complexity theory (ILPS22) proposed the following definition: a randomized algorithm will be replicable if two distinct runs of the algorithm on two sets of samples drawn from the same distribution, with internal randomness fixed between both runs, produces the same output with high probability. A formal definition appears in the upcoming section (see Definition 2.4.7).

The main question arising is whether we can existing Machine Learning algorithms are replicable and, if no, is it possible to design such algorithms? This task is a classical question in Science (Bak16a) but is fairly unexplored in the Theoretical Computer Science community (ILPS22).

Challenge 4. *What is the cost of replicability in the design of ML algorithms?*

We mention that the notion of *cost* could correspond to e.g., the sample and computational overhead of designing replicable algorithms for statistical tasks or the regret incurred by replicable online learners compared to their non-replicable counterparts.

The second notion that we will be interested in is the fundamental notion of differential privacy (for a formal definition, see Definition 2.4.8). As mentioned in (DR14), differential privacy addresses the paradox of learning nothing about an individual while learning useful information about a population. In this thesis, we will formally study the interrelations between replicability, differentialy privacy and other forms of stability. This challenge has been previously raised in (ILPS22; GKM21; CLN$^+$16).

Challenge 5. *Are there formal connections between replicability, differential privacy and other notions of algorithmic* stability*?*

We view both replicability and differential privacy as two fundamental blocks in the area of Responsible Machine Learning. Hence, we believe that establishing formal connections between a priori not clearly related notions of "reliability" is a way to increase our understanding towards the design of responsible Machine Learning systems.

2.3 Summary of Contribution

In this section we summarize the results of this thesis and we present an overview of the structure of the thesis. We start with shortly explaining the content of each one of the chapters. The first half of the thesis deals with Robust Machine Learning, while the second one handles problems related to Responsible Machine Learning.

Robust Machine Learning

In the first part of the thesis (Chapter 3-5), we will deal with questions regarding Robust Machine Learning. In particular, the first three chapters are organized as follows.

Chapter 3 - Truncated Data. We first study the problem of estimating the parameters of a Boolean product distribution in d dimensions, when the samples are truncated by a set $S \subseteq \{0,1\}^d$ accessible through a membership oracle (Challenge 1). We introduce a natural notion of fatness of the truncation set S, under which truncated samples reveal enough information about the true distribution. We show that if the truncation set is sufficiently fat, samples from the true distribution can be generated from truncated samples. A stunning consequence is that virtually any statistical task that can be performed efficiently for Boolean product distributions, can also be performed from truncated samples, with a small increase in sample complexity. Exploring the limits of learning discrete models from truncated samples, we identify natural conditions that are necessary for efficient identifiability. By carefully adapting the Stochastic Gradient Descent approach of (DGTZ18), we show that these conditions are also sufficient for efficient learning of truncated Boolean product distributions.

Chapter 4 - Coarsened Data. In this chapter, we formally define and and study the problem of learning from coarse/partial data (Challenge 2). Instead of observing the actual labels from a set $\mathcal{Z}$, we observe coarse labels corresponding to a partition of $\mathcal{Z}$ (or a mixture of partitions). Our main algorithmic result is that essentially any problem learnable from fine grained labels can also be learned efficiently when the coarse data are sufficiently informative. We obtain our result through a generic reduction for answering Statistical Queries (SQ) over fine grained labels given only coarse labels. The number of coarse labels required depends polynomially on the information distortion due to coarsening and the number of fine labels $|\mathcal{Z}|$. We also adopt an unsupervised perspective to the problem of learning from coarsened data. We investigate a central problem in censored statistics: Gaussian mean estimation from coarsened data. We provide an efficient algorithm when the sets in the partition are convex and establish that the problem is NP-hard even for very simple non-convex sets. On a technical side, our algorithmic result relies on the elegant Brascamp–Lieb inequality and our hardness result is based on (perhaps surprising) connections between coarsened Gaussian distributions and the fundamental problem of finding the maximum cut in a graph, which is known to be NP-hard even to approximate.

Chapter 5 - Corrupted Data. Label Ranking (LR) is the supervised task of learning a sorting function that maps feature vectors $\boldsymbol{x} \in \mathbb{R}^d$ to rankings $\sigma(\boldsymbol{x}) \in \mathbb{S}_k$

over a finite set of k labels. We focus on the fundamental case of learning linear sorting functions (LSFs) under Gaussian marginals: $\boldsymbol{x}$ is sampled from the d-dimensional standard normal and the ground truth ranking $\sigma^\star(\boldsymbol{x})$ is the ordering induced by sorting the coordinates of the vector $\boldsymbol{W}^\star\boldsymbol{x}$, where $\boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$ is unknown. We consider learning LSFs in the presence of bounded noise (Challenge 3): assuming that a noiseless example is of the form $(\boldsymbol{x}, \sigma^\star(\boldsymbol{x}))$, we observe $(\boldsymbol{x}, \pi)$, where for any pair of elements $i \neq j$, the probability that the order of i, j is different in π than in $\sigma^\star(\boldsymbol{x})$ is *at most* $\eta < 1/2$. We design efficient non-proper and proper learning algorithms that learn hypotheses within normalized Kendall's Tau distance ϵ from the ground truth with $N = \widetilde{O}(d \log(k)/\epsilon)$ labeled examples and runtime $\mathrm{poly}(N, k)$. For the more challenging top-r disagreement loss, we give an efficient proper learning algorithm that achieves ϵ top-r disagreement with the ground truth with $N = \widetilde{O}(dkr/\epsilon)$ samples and $\mathrm{poly}(N)$ runtime.

Responsible Machine Learning

In the second part of the thesis (Chapter 6-7), we study problems concerning Responsible Machine Learning. The chapters are organized as follows.

Chapter 6 - Replicable Bandit Algorithm Design. We introduce the notion of replicable policies in the context of stochastic bandits, one of the canonical problems in interactive learning. A policy in the bandit environment is called replicable if it pulls, with high probability, the *exact* same sequence of arms in two different and independent executions (i.e., under independent reward realizations). We show that not only do replicable policies exist, but also they achieve almost the same optimal (non-replicable) regret bounds in terms of the time horizon. More specifically, in the stochastic multi-armed bandits setting, we develop a policy with an optimal problem-dependent regret bound whose dependence on the replicability parameter is also optimal. Similarly, for stochastic linear bandits (with finitely and infinitely many arms) we develop replicable policies that achieve the best-known problem-independent regret bounds with an optimal dependency on the replicability parameter. Our results show that even though randomization is crucial for the exploration-exploitation trade-off, an optimal balance can still be achieved while pulling the exact same arms in two different rounds of executions.

Chapter 7 - Statistical Indistinguishability, Privacy and Replicability. When two different parties use the same learning rule on their own data, how can we test whether the distributions of the two outcomes are similar? In this chapter, we study the similarity of outcomes of learning rules through the lens of the Total Variation (TV) distance of distributions. We say that a learning rule is TV indistinguishable if the expected TV distance between the posterior distributions of its outputs, executed on two training data sets drawn independently from the same distribution, is small. We first investigate the learnability of hypothesis classes using TV indistinguishable learners. Our main results are information-theoretic equivalences between TV indistinguishability and existing algorithmic stability notions such as replicability and approximate differential privacy. Then, we provide statistical amplification and boosting algorithms for TV indistinguishable learners.

2.4 Technical Overview of the Thesis

In this section we delve into a more technical level in the context of this thesis and we provide a summary of the results of each chapter. In each upcoming section, we motivate the problem discussed, provide some preliminary definitions whenever needed and informally state the results of this thesis.

2.4.1 Learning from Truncated Samples - Chapter 3

Parameter estimation and learning from truncated samples is an important and challenging problem in Statistics. The goal is to estimate the parameters of the true distribution based only on samples that fall within a (possibly small) subset S of the distribution's support. Sample truncation occurs naturally in a variety of settings in science, engineering, economics, business and social sciences. Typical examples include selection bias in epidemiology and medical studies, and anecdotal "paradoxes" in damage and injury analysis explained by survivor bias. Statistical estimation from truncated samples goes back to at least (Gal97), who analyzed truncated samples corresponding to speeds of American trotting horses, and includes classical results on the use of the moments method (PL08; Lee14) and the maximum likelihood method (Fis31) for estimating a univariate Gaussian distribution from truncated samples (see also (DGTZ18) for a detailed discussion on the history and the significance of statistical estimation from truncated samples).

In the last few years, there has been an increasing interest in computationally and statistically efficient algorithms for learning multivariate Gaussian distributions from truncated samples (when the truncation set is known (DGTZ18) or unknown (KTZ19)) and for training linear regression on models based on truncated (or censored) data (DGTZ19a). In addition to the elegant and powerful application of Stochastic Gradient Descent to optimizing a seemingly unknown maximum likelihood function from truncated samples, a significant contribution of (DGTZ18; KTZ19; DGTZ19a) concerns necessary conditions for efficient statistical estimation of multivariate Gaussian or regression models from truncated samples. Moreover, (NP19) showed how to use Expectation-Maximization for learning mixtures of two Gaussian distributions from truncated samples.

Despite the strong results above on efficient learning from truncated samples for continuous settings, we are not aware of any previous work on learning discrete models from truncated samples. We note that certain elements of the prior approaches in inference from truncated data are inherently continuous and it is not clear to which extent (and under which conditions) can be adapted to a discrete setting. E.g., statistical estimation from truncated samples in a discrete setting should deal with a situation where the truncation removes virtually all randomness from certain directions, something that cannot be the result of nontrivial truncations in a continuous setting.

Our Contribution on Challenge 1. Motivated by this gap in relevant literature, we investigate efficient parameter estimation of discrete models from truncated samples. We start with the fundamental setting of a Boolean product distribution $\mathcal{D}$ on the d-dimensional hypercube truncated by a set S, which is accessible through membership queries. The marginal of $\mathcal{D}$ in each direction i is an independent Bernoulli distribution with parameter $p_i \in (0,1)$. Our goal is to compute an estimation $\widehat{\boldsymbol{p}}$ of the parameter vector $\boldsymbol{p}$ of $\mathcal{D}$ such that $\|\boldsymbol{p} - \widehat{\boldsymbol{p}}\|_2 \leq \epsilon$, with probability of at least $1-\delta$, with time and sample complexity polynomial in d, $1/\epsilon$ and $\log(1/\delta)$. We note that such an estimation $\widehat{\boldsymbol{p}}$ (or an estimation $\widehat{\boldsymbol{z}}$ of the logit parameters $\boldsymbol{z} = (\log \frac{p_1}{1-p_1}, \dots, \log \frac{p_d}{1-p_d})$ of similar accuracy) implies an estimation of the true distribution within total variation distance ϵ.

Significantly departing from the maximum likelihood estimation approach of (DGTZ18; KTZ19; DGTZ19a), we introduce a natural notion of fatness of the truncation set S, under which samples from the truncated distribution $\mathcal{D}_S$ reveal enough information about the true distribution $\mathcal{D}$. Roughly speaking, a truncated Boolean product distribution $\mathcal{D}_S$ is α*-fat* in some direction i of the Boolean hypercube, if for an α probability mass of the truncated samples, the neighboring sample with its i-th coordinate flipped is also in S. Therefore, with probability α, conditional on the remaining coordinates, the i-th coordinate of a sample is distributed as the marginal of the true distribution $\mathcal{D}$ in direction i. So, if the truncated distribution $\mathcal{D}_S$ is α-fat in all directions (e.g., the halfspace of all vectors with L_1 norm at most k is a fat subset of the Boolean hypercube), a sample from $\mathcal{D}_S$ is quite likely to reveal significant information about the true distribution $\mathcal{D}$. Building on this intuition, we show how samples from the true distribution $\mathcal{D}$ can be generated from few truncated samples (see also Algorithm 1):

Informal Theorem. *With an expected number of $O(\log(d)/\alpha)$ samples from the α-fat truncation of a Boolean product distribution $\mathcal{D}$, we can generate a sample $\boldsymbol{x} \in \{0,1\}^d$ distributed as in $\mathcal{D}$.*

We show (Lemma 3.2.2) that fatness is also a necessary condition for the above result. A stunning consequence of our procedure is that virtually any statistical task (e.g., learning in total variation distance, parameter estimation, sparse recovery, uniformity or identity testing) that can be performed efficiently for a Boolean product distribution $\mathcal{D}$, can also be performed using truncated samples from $\mathcal{D}$, at the expense of a factor $O(\log(d)/\alpha)$ increase in time and sample complexity. In

Section 3.2, we obtain, as simple corollaries, that the statistical tasks described in (ADK15; DKS17a; CDKS17; CKM$^+$19) for Boolean product distributions can be performed using only truncated samples!

To further demonstrate the power and the wide applicability of our approach, we extend the notion of fatness to the richer and more complex setting of ranking distributions on d alternatives. In Section 3.2.5, we show how to implement efficient statistical inference of Mallows models using samples from a fat truncated Mallows distribution (see Theorem 3.2.10).

Natural and powerful though, fatness is far from being necessary for efficient parameter estimation from truncated samples. Seeking a deeper understanding of the challenges of learning discrete models from truncated samples, we identify, in Section 3.3, three natural conditions that we show to be necessary for efficient parameter estimation in our setting:

Assumption 1: The support of the distribution $\mathcal{D}$ on S should be rich enough, in the sense that its truncation $\mathcal{D}_S$ should assign positive probability to a $\boldsymbol{x}^\star \in S$ and d other vectors that remain linearly independent after we subtract $\boldsymbol{x}^\star$ from them.

Assumption 2: S is accessible through a membership oracle that reveals whether $\boldsymbol{x} \in S$, for any $\boldsymbol{x}$ in the d-dimensional hypercube.

Assumption 3: The truncation of $\mathcal{D}$ by S leaves enough randomness in all directions. More precisely, we require that in any direction $\boldsymbol{w} \in \mathbb{R}^d$, any two samples from the truncated distribution $\mathcal{D}_S$ have sufficiently different projections on $\boldsymbol{w}$, with non-negligible probability.

Assumption 2 ensures that the learning algorithm has enough information about S and is also required in the continuous setting. Without oracle access to S, for any Boolean product distribution $\mathcal{D}$, we can construct a (possibly exponentially large) truncation set S such that sampling from the truncated distribution $\mathcal{D}_S$ appears identical to sampling from the uniform distribution, until the first duplicate sample appears (our construction is similar to (DGTZ18, Lemma 12)).

Similarly to (DGTZ18), Assumption 2 is complemented by the additional natural requirement that the true distribution $\mathcal{D}$ should assign non-negligible probability mass to the truncation set S (Assumption 4). The reason is that the parameter estimation algorithm evaluates the quality of its current estimation by generating samples in S and comparing them with samples from $\mathcal{D}_S$. Assumptions 2 and 4 ensure that this can be performed efficiently.

Assumptions 1 and 3 are specific to the discrete setting of the Boolean hypercube. Assumption 1 requires that we should be able to normalize the truncation set S, by subtracting a vector $\boldsymbol{x}^\star$, so that its dimension remains d. If this is true, we can recover the parameters of a Boolean product distribution $\mathcal{D}$ from truncated samples by solving a linear system with d equations and d unknowns, which we obtain after normalization. We prove, in Lemma 3.3.1, that Assumption 1 is both sufficient and necessary for parameter recovery from truncated samples in our setting.

Assumption 3 is a stronger version of Assumption 1 and is necessary for efficient parameter estimation from truncated samples in the Boolean hypercube. It essentially requires that with sufficiently high probability, any set X of polynomially many samples from $\mathcal{D}_S$ can be normalized, subtracting a vector $\boldsymbol{x}^\star$, so that X includes a well-conditioned $d \times d$ matrix, after normalization.

Beyond showing that these assumptions are necessary for efficient identifiability, we show that they are also sufficient and provide a computationally efficient algorithm for learning Boolean product distributions. Our algorithm is based on a careful adaptation of the approach of (DGTZ18) which uses Stochastic Gradient Descent on the negative log-likelihood. While the analysis consists of the same conceptual steps as that of (DGTZ18), it requires dealing with a number of technical details that arise due to discreteness. One technical contribution of our work is using the necessary assumptions for identifiability to establish strong-convexity of the negative log-likelihood in a small ball around the true parameters (see Lemma 3.5.12 and Lemma 3.5.9). Our main result is that:

Informal Theorem. *Under Assumptions (1) - (4), Algorithm 4 computes an estimation $\widehat{\boldsymbol{z}}$ of the logit vector $\boldsymbol{z}$ of the true distribution $\mathcal{D}$ such that $\|\boldsymbol{z}-\widehat{\boldsymbol{z}}\|_2 \leq \epsilon$ with probability at least $1-\delta$, and achieves time and sample complexity polynomial in d, $1/\epsilon$ and $\log(1/\delta)$.*

2.4.2 Learning from Coarse Labels - Chapter 4

The most classical problem in Machine Learning and Statistics is that of classification: given labeled examples, the goal is to train some model to achieve low classification error on new potentially unseen examples. In most modern applications, where we train complicated models such as neural nets, large amounts of labeled examples are required. Large datasets such as Imagenet, (RDS$^+$15), often contain thousands of different categories such as animals, vehicles, etc., each one of those containing many *fine grained* subcategories: animals may contain dogs and cats and dogs may be further split into different breeds et cetera.

In the last few years, there have been many works that focus on fine grained recognition, (GLB$^+$18; CDCM18; TSD$^+$20; QCJ$^+$20; LGW17; JLYW19; JLL$^+$20; BSS$^+$20; TKD$^+$19). Collecting a sufficient amount of accurately labeled training examples is a hard and expensive task that often requires hiring experts to annotate the examples. This has motivated the problem of learning from *coarsely* labeled datasets, where a dataset is not fully annotated with fine grained labels but a combination of fine, e.g., cat, and coarse labels, e.g., animal, is given, (DKFF13; RGGV15).

Even though the problem of learning from coarsely labeled data has attracted significant attention from the applied community, from a theoretical perspective little is known.

Our Contribution on Challenge 2. We model coarse labels as subsets of the domain of all possible fine labels. For example, assume that we hire an expert on dog breeds and an expert on cat breeds to annotate a dataset containing images of dogs and cats. With probability $1/2$, we get samples labeled by the dog expert, i.e., labeled according to the partition

$$\{\text{cat} = \{\text{persian cat}, \text{bengal cat}, \ldots\}, \{\text{maltese dog}\}, \{\text{husky dog}\}, \ldots\}\,.$$

On the other hand, the cat expert will provide a fine grained partition over cat breeds and will group together all dog breeds. Our coarse data model captures exactly this mixture of different label partitions.

Definition 2.4.1 (Generative Process of Coarse Data with Context)**.** *Let $\mathcal{X}$ be an arbitrary domain, and let $\mathcal{Z} = \{1, \ldots, k\}$ be the discrete domain of all possible fine labels.*

We generate coarsely labeled examples as follows:

1. *Draw a finely labeled example (x, z) from a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Z}$.*
2. *Draw a coarsening partition $\mathcal{S}$ (of $\mathcal{Z}$) from a distribution π.*
3. *Find the unique set $S \in \mathcal{S}$ that contains the fine label z.*
4. *Observe the coarsely labeled example (x, S).*

We denote $\mathcal{D}_\pi$ the distribution of the coarsely labeled example (x, S).

Our objective motivated by real-world applications can be summarized as follows.

Question 1. *Can we train a model, using coarsely labeled examples $(x, S) \sim \mathcal{D}_\pi$, that classifies finely labeled examples $(x, z) \sim \mathcal{D}$ with accuracy comparable to that of a classifier that was trained on examples with fine grained labels?*

Definition 2.4.1 does not impose any restrictions on the distribution over partitions π. It is clear that if partitions are very rough, e.g., we split $\mathcal{Z}$ into two large disjoint subsets, we lose information about the fine labels and we cannot hope to train a classifier that performs well over finely labeled examples.

In order for Question 1 to be information theoretically possible, we need to assume that the partition distribution π preserves fine-label information. The following definition quantifies this by stating that reasonable partition distributions π are those that preserve the total variation distance between different distributions supported on the domain of the fine labels $\mathcal{Z}$. Let us introduce the notion of *information preserving distributions*: For some $\alpha \in (0,1]$, we will say that the distribution π is an **α-information preserving partition distribution** if for every two distributions $\mathcal{D}^1, \mathcal{D}^2$ over the domain $\mathcal{Z}$, it holds that $d_{\mathrm{TV}}(\mathcal{D}^1_\pi, \mathcal{D}^2_\pi) \geq \alpha \cdot d_{\mathrm{TV}}(\mathcal{D}^1, \mathcal{D}^2)$, where $d_{\mathrm{TV}}(\mathcal{D}^1, \mathcal{D}^2)$ is the total variation distance of $\mathcal{D}^1$ and $\mathcal{D}^2$. Intuitively, this definition captures the distortion that the coarsening provokes to any pair of probability measures.

For example, the partition distribution defined in the dog/cat dataset scenario, discussed before Definition 2.4.1, is 1/2-information preserving, since we observe fine labels with probability 1/2. In this case, it is easy, at the expense of losing the statistical power of the coarse labels, to combine the finely labeled examples from both experts in order to obtain a dataset consisting only of fine labels. However, our model allows the partitions to have arbitrarily complex combinatorial structure that makes the process of “inverting" the partition transformation computationally challenging. For example, specific fine labels may be complicated functions of coarse labels: “medium sized" and “pointy ears" and “blue eyes" may be mapped to the “husky dog" fine label.

Our first result is a positive answer to Question 1 in essentially full generality: We show that concept classes that are efficiently learnable in the Statistical Query (SQ) model of (Kea98), are also learnable from coarsely labeled examples. Our result is similar in spirit with the result of (Kea98), where it is proved that SQ learnability implies learnability under random classification noise. Hence, we provide a generic reduction and so that we can efficiently compute statistical queries over fine labels provided sample access to coarsely labeled examples.

Informal Theorem (SQ Learnability implies Learnability from Coarse Examples)**.** *Any concept class $\mathcal{C} \subseteq [k]^{\mathcal{X}}$ that is efficiently learnable with M statistical queries from finely labeled examples $(x,z) \sim \mathcal{D}$, can be efficiently learned from $O(\mathrm{poly}(k/\alpha)) \cdot M$ coarsely labeled examples $(x,S) \sim \mathcal{D}_\pi$ under any α-information preserving partition distribution π.*

Statistical Queries are queries of the form $\mathbb{E}_{(x,z)\sim\mathcal{D}}[q(x,z)]$ for some query function $q(x,z)$. It is known that almost all known machine learning algorithms (AD98; BFKV98; BDMN05; DV08; BF15; FGR$^+$17) can be implemented in the SQ model. In particular, in (FGV17), it is shown that (Stochastic) Gradient Descent can be simulated by statistical queries. This implies that our result can be applied, even in cases where it is not possible to obtain formal optimality guarantees, e.g., training deep neural nets. We can train such models using coarsely labeled data and guarantee the same performance as if we had direct access to fine labels (see also Section 4.1.3). [1] As another application, we consider the problem of multiclass logistic regression with coarse labels. It is known, see e.g., (FHT01), that

[1] Given any objective of the form $L(\boldsymbol{v}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\ell(\boldsymbol{v};\boldsymbol{x},y)]$, its gradients correspond to $\nabla_{\boldsymbol{v}} L(\boldsymbol{v}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\nabla_{\boldsymbol{v}}\ell(\boldsymbol{v};\boldsymbol{x},y)]$. Having Statistical Query access to the distribution of (x,y), we can directly obtain estimates of the above gradients using the query func-

given finely labeled examples $(x, z) \sim \mathcal{D}$, the likelihood objective for multiclass logistic regression is concave with respect to the weight matrix. Even though the likelihood objective is no-longer concave when we consider coarsely labeled examples $(x, S) \sim \mathcal{D}_\pi$, our theorem bypasses this difficulty and allows us to efficiently perform multiclass logistic regression with coarse labels.

Inference from coarse data naturally arises also in unsupervised, i.e., distribution learning settings: instead of directly observing samples from the target distribution, we observe "representative" points that correspond to larger sets of samples. For example, instead of observing samples from a real valued random variable, we round them to the closest integer. An important unsupervised problem that fits in the coarse data framework is censored statistics, (Coh16; Wol79; Bre96; Sch86). Interval censoring, that arises in insurance adjustment applications, corresponds to observing points in some interval and point masses at the endpoints of the interval instead of observing fine grained data from the whole real line. Moreover, the problem of learning the distribution of the output of neural networks with non-smooth activations (e.g., ReLU networks, (WDS19)) also fits in our model of distribution learning with coarse data, see Figure 2.1(c).

We make progress towards the direction of learning parametric distributions from coarse samples. In many important applications, instead of a discrete distribution over fine labels, a continuous parametric model is used. A popular example is when the domain $\mathcal{Z}$ of Definition 2.4.1 is the entire Euclidean space $\mathbb{R}^d$, and the distribution of finely labeled examples is a Gaussian distribution whose parameters possibly depend on the context x. Such censored regression settings are known as Tobit models (Tob58; Mad86; Gou00). Lately, significant progress has been made from a computational point of view in such censored/truncated settings in the distribution specific setting, e.g., when the underlying distribution is Gaussian (DGTZ18; KTZ19), mixtures of Gaussians (NP19), linear regression (DGTZ19b; IZD20; DRZ20). In this distribution specific setting, we consider the most fundamental problem of learning the mean of a Gaussian distribution given coarse data.

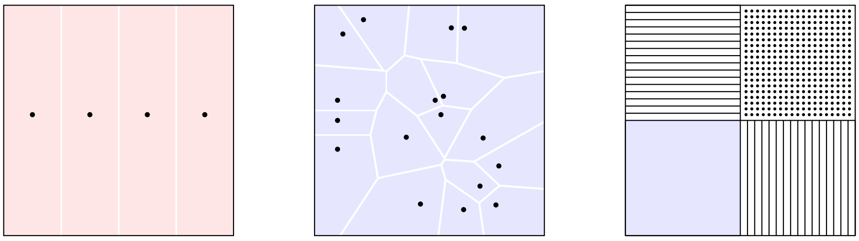
Definition 2.4.2 (Coarse Gaussian Data). *Consider the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^\star)$, with mean $\boldsymbol{\mu}^\star \in \mathbb{R}^d$ and identity covariance matrix. We generate a sample as follows:*

1. *Draw $\boldsymbol{z}$ from $\mathcal{N}(\boldsymbol{\mu}^\star)$.*
2. *Draw a partition $\mathcal{S}$ (of $\mathbb{R}^d$) from π.*
3. *Observe the set $S \in \mathcal{S}$ that contains $\boldsymbol{z}$.*

We denote the distribution of S as $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$.

tions $q_i(\boldsymbol{x}, y) = (\nabla_{\boldsymbol{v}} \ell(\boldsymbol{v}; \boldsymbol{x}, y))_i$. In (FGV17), the precise accuracy required for specific SQ implementations of first order methods depends on the complexity of the underlying distribution and the particular objective function $\ell(\cdot)$.

We first study the above problem, from a computational viewpoint. For the corresponding problems in censored and truncated statistics no geometric assumptions are required for the sets: In (DGTZ18), it was shown that an efficient algorithm exists for arbitrarily complex truncation sets. In contrast in our more general model of coarse data we show that having sets with geometric structure is necessary. In particular we require that every set of the partition is convex, see Figure 2.1(b,c). We show that when the convexity assumption is dropped, learning from coarse data is a computationally hard problem even under a mixture of very simple sets.



(a) Non-Identifiable (b) Convex Partition (c) ReLU Case

Figure 2.1: (a) is a very rough partition, that makes learning the mean impossible: Gaussians $\mathcal{N}((0,z))$ centered along the same vertical line $(0,z)$ assign exactly the same probability to all cells of the partitions and therefore, $d_{\mathrm{TV}}(\mathcal{N}_\pi((0,z_1)),\mathcal{N}_\pi((0,z_2)))=0$: it is impossible to learn the second coordinate of the mean. (b) is a convex partition of $\mathbb{R}^2$, that makes recovering the Gaussian possible. (c) is the convex partition corresponding to the output distribution of one layer ReLU networks. When both coordinates are positive, we observe a fine sample (black points correspond to singleton sets). When exactly one coordinate (say $\boldsymbol{x}_1$) is positive, we observe the line $L_z=\{\boldsymbol{x}:\boldsymbol{x}_2<0,\boldsymbol{x}_1=z>0\}$ that corresponds to the ReLU output $(\boldsymbol{x}_1,0)$. If both coordinates are negative, we observe the set $\{\boldsymbol{x}:\boldsymbol{x}_1<0,\boldsymbol{x}_2<0\}$, that corresponds to the point $(0,0)$.

Informal Theorem. *Let π be a general partition distribution. Unless* RP = NP*, no algorithm with sample access to $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, can compute, in* $\mathrm{poly}(d)$ *time, a $\widetilde{\boldsymbol{\mu}}\in\mathbb{R}^d$ such that $d_{\mathrm{TV}}(\mathcal{N}_\pi(\widetilde{\boldsymbol{\mu}}),\mathcal{N}_\pi(\boldsymbol{\mu}^\star))<1/d^c$ for some absolute constant $c>1$.*

We prove our hardness result using a reduction from the well known Max-Cut problem, which is known to be NP-hard, even to approximate (Hås01). In our reduction, we use partitions that consist of simple sets: fat hyperplanes, ellipsoids and their complements: the computational hardness of this problem is rather inherent and not due to overly complicated sets.

On the positive side, we identify a geometric property that enables us to design a computationally efficient algorithm for this problem: Namely we require all the sets of the partitions to be *convex*, e.g., Figure 2.1(b,c). We remark that having finite or countable subsets, is not a requirement of our model. For example, we can handle convex partitions of the form (c) that correspond to the output distribution of a ReLU neural network, see (WDS19). We continue with our theorem for learning Gaussians from coarse data.

Informal Theorem. *Let $\epsilon \in (0,1)$. Consider the generative process of coarse d-dimensional Gaussian data $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$. Assume that the partition distribution π is α-information preserving and is supported on convex partitions of $\mathbb{R}^d$. There exists an algorithm that draws $N = \widetilde{O}(d/(\epsilon^2\alpha^2))$ samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$ and computes an estimate $\widetilde{\boldsymbol{\mu}}$ that satisfies $d_{\mathrm{TV}}(\mathcal{N}(\widetilde{\boldsymbol{\mu}}),\mathcal{N}(\boldsymbol{\mu}^\star)) \le \epsilon$, with probability at least* 99%.

In the above, the definition of information preservation is very similar with the one given in Definition 4.1.2. We postpone the details for the associated chapter. Our algorithm for mean estimation of a Gaussian distribution relies on the log-likelihood being concave when the partitions are convex. We remark that, similar to our approach, one can use the concavity of likelihood to get efficient algorithms for regression settings, e.g., Tobit models, where the mean of the Gaussian is given by a linear function of the context $\boldsymbol{A}\boldsymbol{x}$ for some unknown matrix $\boldsymbol{A}$.

2.4.3 Learning with Bounded Noise - Chapter 5

Label Ranking (LR) is the problem of learning a hypothesis that maps features to rankings over a finite set of labels. Given a feature vector $\boldsymbol{x} \in \mathbb{R}^d$, a sorting function $\sigma(\cdot)$ maps it to a ranking of k alternatives, i.e., $\sigma(\boldsymbol{x})$ is an element of the symmetric group with k elements, $\mathbb{S}_k$. Assuming access to a training dataset of features labeled with their corresponding rankings, i.e., pairs of the form $(\boldsymbol{x}, \pi) \in \mathbb{R}^d \times \mathbb{S}_k$, the goal of the learner is to find a sorting function $h(\boldsymbol{x})$ that generalizes well over a fresh sample. LR has received significant attention over the years (DSM03; SS07; HFCB08; CH08; FHMB08) due to the large number of applications. For example, ad targeting (DGR+14) is an LR instance where for each user we want to use their feature vector to predict a ranking over ad categories and present them with the most relevant. The practical significance of LR has lead to the development of many techniques based on probabilistic models and instance-based methods (CH08; CDH10), (GDV12; ZLGQ14), decision trees (CHH09), entropy-based ranking trees (RdSRSK15), bagging (AGM17), and random forests (dSSKC17; ZQ18). However, almost all of these works come without provable guarantees and/or fail to learn in the presence of noise in the observed rankings.

Linear Sorting Functions (LSFs). We focus on the fundamental concept class of Linear Sorting functions (HPRZ03). A linear sorting function parameterized by a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ with k rows $\boldsymbol{W}_1, \dots, \boldsymbol{W}_k$ takes a feature $\boldsymbol{x} \in \mathbb{R}^d$, maps it to $\boldsymbol{W}\boldsymbol{x} = (\boldsymbol{W}_1 \cdot \boldsymbol{x}, \dots, \boldsymbol{W}_k \cdot \boldsymbol{x}) \in \mathbb{R}^k$ and then outputs an ordering $(i_1, \dots, i_k)$ of the k alternatives such that $\boldsymbol{W}_{i_1} \cdot \boldsymbol{x} \geq \boldsymbol{W}_{i_2} \cdot \boldsymbol{x} \geq \dots \geq \boldsymbol{W}_{i_k} \cdot \boldsymbol{x}$. In other words, a linear sorting function ranks the k alternatives (corresponding to rows of $\boldsymbol{W}$) with respect to how well they correlate with the feature $\boldsymbol{x}$. We denote a linear sorting function with parameter $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ by $\sigma_{\boldsymbol{W}}(\boldsymbol{x}) \triangleq \mathrm{argsort}(\boldsymbol{W}\boldsymbol{x})$ where $\mathrm{argsort} : \mathbb{R}^k \to \mathbb{S}_k$ takes as input a vector $(v_1, \dots, v_k) \in \mathbb{R}^k$, sorts it in decreasing order to obtain $v_{i_1} \geq v_{i_2} \geq \dots \geq v_{i_k}$ and returns the ordering $(i_1, \dots, i_k)$.

Noisy Ranking Distributions. Learning LSFs in the noiseless setting can be done efficiently by using linear programming. However, the common assumption both in theoretical and in applied works is that the observed rankings are noisy in the sense that they do not always correspond to the ground-truth ranking. We assume that the probability that the order of two elements i, j in the observed ranking π is different than their order in the ground-truth ranking $\sigma^\star$ is at most $\eta < 1/2$.

Definition 2.4.3 (Noisy Ranking Distribution). *Fix $\eta \in [0, 1/2)$. An η-**noisy ranking distribution** $\mathcal{M}(\sigma^\star)$ with ground-truth ranking $\sigma^\star \in \mathbb{S}_k$ is a probability measure over $\mathbb{S}_k$ that, for any $i, j \in [k]$, with $i \neq j$, satisfies $\mathbf{Pr}_{\pi \sim \mathcal{M}(\sigma^\star)}[i \prec_\pi j \mid i \succ_{\sigma^\star} j] \leq \eta$.* [2]

Note that, when $\eta = 0$, we always observe the ground-truth permutation and, in the case of $\eta = 1/2$, we may observe a uniformly random permutation. We remark that most natural ranking distributions satisfy this bounded noise property, e.g., (i) the Mallows model, which is probably the most fundamental ranking distribution (see, e.g., (BM09; LB11; CPS13; ABSV14; BFFSZ19; FKS21; DOS18; LM18; MW20; LM21) for a small sample of this line of research) and (ii) the Bradley-Terry-Mallows model (Mal57), which corresponds to the ranking distribution analogue of the Bradley-Terry-Luce model (BT52b; Luc12) (the most studied pairwise comparisons model; see, e.g., (Hun04; NOS17; APA18) and the references therein). For more details, see Appendix 5.10.

We consider the fundamental setting where the feature vector $\boldsymbol{x} \in \mathbb{R}^d$ is generated by a standard normal distribution and the ground-truth ranking for each sample $\boldsymbol{x}$ is given by the LSF $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$ for some unknown parameter matrix $\boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$. For a fixed $\boldsymbol{x}$, the ranking that we observe comes from an η-noisy ranking distribution with ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$.

Definition 2.4.4 (Noisy Linear Label Ranking Distribution). *Fix $\eta \in [0, 1/2)$ and some ground-truth parameter matrix $\boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$. We assume that the η-**noisy linear label ranking distribution** $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{S}_k$ satisfies the following:*

[2] We use $i \succ_\pi j$ (resp. $i \prec_\pi j$) to denote that the element i is ranked higher (resp. lower) than j according to the ranking π.

1. *The $\boldsymbol{x}$-marginal of $\mathcal{D}$ is the d-dimensional standard normal distribution.*
2. *For any $(\boldsymbol{x}, \pi) \sim \mathcal{D}$, the distribution of π conditional on $\boldsymbol{x}$ is an η-noisy ranking distribution with ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$.*

At first sight, the assumption that the underlying $\boldsymbol{x}$-marginal is the standard normal may look too strong. However, for $k = 2$, Definition 5.1.1 captures the problem of learning linear threshold functions with Massart noise. Without assumptions for the $\boldsymbol{x}$-marginal, it is known (DGT19b; CKMY20; DK20; NT22) that optimal learning of halfspaces under Massart noise requires super-polynomial time (in the Statistical Query model of (Kea98)). On the other hand, a lot of recent works (BZ17; MV19; DKTZ20; ZSA20; ZL21) have obtained efficient algorithms for learning Massart halfspaces under Gaussian marginals. The goal of this work is to provide efficient algorithms for the more general problem of learning LSFs with bounded noise under Gaussian marginals.

Our Contribution on Challenge 3. The main contributions of this chapter are the first efficient algorithms for learning LSFs with bounded noise with respect to Kendall's Tau distance and top-r disagreement loss.

Learning in Kendall's Tau Distance

The most standard metric in rankings (SSBD14) is Kendall's Tau (KT) distance which, for two rankings $\pi, \tau \in \mathbb{S}_k$, measures the fraction of pairs (i, j) on which they disagree. That is, $\Delta_{\mathrm{KT}}(\pi, \tau) = \sum_{i \prec_\pi j} \mathbf{1}\{i \succ_\tau j\}/\binom{k}{2}$. Our first result is an efficient learning algorithm that, given samples from an η-noisy linear label ranking distribution $\mathcal{D}$, computes a parameter matrix $\boldsymbol{W}$ that ranks the alternatives almost optimally with respect to the KT distance from the ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\cdot)$.

Informal Theorem. *Fix $\eta \in [0, 1/2)$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1 with ground-truth LSF $\sigma_{\boldsymbol{W}^\star}(\cdot)$. There exists an algorithm that draws $N = \widetilde{O}\left(\frac{d}{\epsilon(1-2\eta)^6} \log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in sample-polynomial time, and computes a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ such that, with probability at least $1 - \delta$,*

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} [\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq \epsilon\,.$$

This result gives the first efficient algorithm with provable guarantees for the supervised problem of learning noisy linear rankings. We remark that the sample complexity of our learning algorithm is qualitatively optimal (up to logarithmic factors) since, for $k = 2$, our problem subsumes learning a linear classifier with

Massart noise [3] for which $\Omega(d/\epsilon)$ are known to be information theoretically necessary (MN06). Moreover, our learning algorithm is *proper* in the sense that it computes a linear sorting function $\sigma_{\boldsymbol{W}}(\cdot)$. As opposed to improper learners (see also **??**), a proper learning algorithm gives us a compact representation (storing $\boldsymbol{W}$ requires $O(kd)$ memory) of the sorting function that allows us to efficiently compute (with runtime $O(kd + k \log k)$) the ranking corresponding to a fresh datapoint $\boldsymbol{x} \in \mathbb{R}^d$.

Learning in top-r Disagreement

We next present our learning algorithm for the top-r metric formally defined as $\Delta_{\mathrm{top}-r}(\pi, \tau) = \mathbf{1}\{\pi_{1..r} \neq \tau_{1..r}\}$, where by $\pi_{1..r}$ we denote the ordering on the first r elements of the permutation π. The top-r metric is a disagreement metric in the sense that it takes binary values and for $r = 1$ captures the standard (multiclass) top-1 classification loss. We remark that, in contrast with the top-r classification loss, which only requires the predicted label to be in the top-r predictions of the model, the top-r ranking metric that we consider here requires that the model puts *the same elements in the same order* as the ground truth in the top-r positions. The top-r ranking is well-motivated as, for example, in ad targeting (discussed in **??**) we want to be accurate on the top-r ad categories for a user so that we can diversify the content that they receive.

Informal Theorem. *Fix $\eta \in [0, 1/2)$, $r \in [k]$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1 with ground-truth LSF $\sigma_{\boldsymbol{W}^\star}(\cdot)$. There exists an algorithm that draws $N = \widetilde{O}\left(\frac{drk}{\epsilon(1-2\eta)^6} \log(1/\delta)\right)$ samples from $\mathcal{D}$, runs in sample-polynomial time and computes a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ such that, with probability at least $1 - \delta$,*

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} [\Delta_{\mathrm{top}-r}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq \epsilon\,.$$

As a direct corollary of our result, we obtain a proper algorithm for learning the top-1 element with respect to the standard 0-1 loss that uses $\widetilde{O}(kd)$ samples. In fact, for small values of r, i.e., $r = O(1)$, our sample complexity is essentially tight. It is known that $\Theta(kd)$ samples are information theoretically necessary (Nat89) for top-1 classification. [4] For the case $r = k$, i.e., when we want to learn the whole ranking with respect to the 0-1 loss, our sample complexity is $O(k^2 d)$. However, using arguments similar to (DSBDSS11), one can show that in fact $O(dk)$ ranking

[3] Notice that in this case Kendall's Tau distance is simply the standard 0-1 binary loss.

[4] Strictly speaking, those lower bounds do not directly apply in our setting because our labels are whole rankings instead of just the top classes but, in the Appendix 5.9, we show that we can adapt the lower bound technique of (DSBDSS11) to obtain the same sample complexity lower bound for our ranking setting.

samples are sufficient in order to learn the whole ranking with respect to the 0-1 loss. In this case, it is unclear whether a better sample complexity can be achieved with an efficient algorithm and we leave this as an interesting open question for future work.

2.4.4 Replicable Bandit Algorithms - Chapter 6

The notion of a replicable algorithm in the context of (offline) learning was proposed by (ILPS22), where it is shown how any statistical query algorithm can be made replicable with a moderate increase in its sample complexity. Using this result, they provide replicable algorithms for finding approximate heavy-hitters, medians, and the learning of half-spaces. Reproducibility has been also considered in the context of optimization by (AJJ+22). We mention that in (AJJ+22) the notion of a replicable algorithm is different from our work and that of (ILPS22), in the sense that the outputs of two different executions of the algorithm do not need to be exactly the same. From a more application-oriented perspective, (SL22) study irreproducibility in recommendation systems and propose the use of smooth activations (instead of ReLUs) to improve recommendation reproducibility. In general, the reproducibility crisis is reported in various scientific disciplines (Ioa05; McN14; Bak16b; GFI16; LKM+18; HIB+18). For more details we refer to the report of the NeurIPS 2019 Reproducibility Program (PVLS+21) and the ICLR 2019 Reproducibility Challenge (PSF+19).

We initiate the study of replicability in the bandit setting. A multi-armed bandit (MAB) is a one-player game that is played over T rounds where there is a set of different arms/actions $\mathcal{A}$ of size $|\mathcal{A}| = K$ (in the more general case of linear bandits, we can consider even an infinite number of arms). In each round $t = 1, 2, \ldots, T$, the player pulls an arm $a_t \in \mathcal{A}$ and receives a corresponding reward r_t. In the stochastic setting, the rewards of each arm are sampled in each round independently, from some fixed but unknown, distribution supported on $[0, 1]$. Crucially, each arm has a potentially different reward distribution, but the distribution of each arm is fixed over time. A bandit algorithm $\mathbb{A}$ at every round t takes as input the sequence of arm-reward pairs that it has seen so far, i.e., $(a_1, r_1), \ldots, (a_{t-1}, r_{t-1})$, then uses (potentially) some internal randomness ξ to pull an arm $a_t \in \mathcal{A}$ and, finally, observes the associated reward $r_t \sim \mathcal{D}_{a_t}$.

Our Contribution on Challenge 4. We propose the following natural notion of a replicable bandit algorithm, which is inspired by the definition of (ILPS22)[5]. Intuitively, a bandit algorithm is replicable if two distinct executions of the algorithm, with internal randomness fixed between both runs, but with independent reward realizations, give the exact same sequence of played arms, with high probability. More formally, we have the following definition.

[5] Initially, this property was called “reproducibility”, but it was later pointed that the correct term is “replicability”.

Definition 2.4.5 (Replicable Bandit Algorithm)**.** *Let $\rho \in [0,1]$. We call a bandit algorithm $\mathbb{A}$ ρ-replicable in the stochastic setting if for any distribution $\mathcal{D}_{a_j}$ over $[0,1]$ of the rewards of the j-th arm $a_j \in \mathcal{A}$, and for any two executions of $\mathbb{A}$, where the internal randomness ξ is shared across the executions, it holds that*

$$\mathop{\mathbf{Pr}}_{\xi, \boldsymbol{r}^{(1)}, \boldsymbol{r}^{(2)}} \left[\left(a_1^{(1)}, \ldots, a_T^{(1)} \right) = \left(a_1^{(2)}, \ldots, a_T^{(2)} \right) \right] \geq 1 - \rho \, .$$

Here, $a_t^{(i)} = \mathbb{A}(a_1^{(i)}, r_1^{(i)}, ..., a_{t-1}^{(i)}, r_{t-1}^{(i)}; \xi)$ is the t-th action taken by the algorithm $\mathbb{A}$ in execution $i \in \{1, 2\}$.

We remark that replicable algorithms are predictable, a property which is very desirable when it comes to deploying them in practical applications. In theoretical computer science it is very convenient for algorithm designers to use randomness. However, policy makers are hesitant to use decision-making algorithms whose behavior is brittle and depends heavily on the stochasticity of the environment and its own randomness. The reason why we allow for some fixed internal randomness is that the algorithm designer has control over it, e.g., they can use the same seed for their (pseudo-)random generator between two executions. Clearly, naively designing a replicable bandit algorithm is not quite challenging. For instance, an algorithm that always pulls the same arm or an algorithm that plays the arms in a particular random sequence determined by the shared random seed ξ are both replicable. The caveat is that the performance of these algorithms in terms of expected regret will be quite poor. We aim to design bandit algorithms which are replicable and enjoy small expected regret. In the stochastic setting, the (expected) regret after T rounds is defined as

$$\mathbb{E}[R_T] = T \max_{a \in \mathcal{A}} \mu_a - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{a_t} \right] ,$$

where $\mu_a = \mathbb{E}_{r \sim \mathcal{D}_a}[r]$ is the mean reward for arm $a \in \mathcal{A}$. In a similar manner, we can define the regret in the more general setting of linear bandits (see, Section 6.4). Hence, the overarching question in this chapter is the following:

Is it possible to design replicable bandit algorithms with small expected regret?

At a first glance, one might think that this is not possible, since it looks like replicability contradicts the exploratory behavior that a bandit algorithm should possess. However, our main results answer this question in the affirmative.

In particular, we show the existence of the following bandit algorithms which are provably replicable and enjoy small expected regret for the settings of (i) stochastic multi-armed bandits with K arms, (ii) stochastic linear bandits with K arms and ambient dimension d, and, (iii) stochastic linear bandits with infinitely many arms and ambient dimension d.

Informal Theorem. *Let $\rho \in (0,1), T \in \mathbb{N}$ and $H_\Delta = \sum_{j:\Delta_j>0} 1/\Delta_j$, where Δ_j is the difference between the mean of action j and the optimal action.*

1. *There exists a ρ-replicable algorithm for the stochastic MAB setting with K arms with expected regret*
$$\widetilde{O}(K^2 \log(T) H_\Delta/\rho^2)\,.$$
2. *There exists a ρ-replicable algorithm for the stochastic d-dimensional linear bandit setting with K arms with expected regret*
$$\widetilde{O}(K^2\sqrt{dT}/\rho^2)\,.$$
3. *There exists a ρ-replicable algorithm for the stochastic d-dimensional linear bandit setting with infinite action space with expected regret*
$$\widetilde{O}(\mathrm{poly}(d)\sqrt{T}/\rho^2)\,.$$

2.4.5 Statistically Indistinguishable Learning Algorithms - Chapter 7

Reproducibility of outcomes in scientific research is a necessary condition to ensure that the conclusions of the studies reflect inherent properties of the underlying population and are not an artifact of the methods that scientists used or the random sample of the population that the study was conducted on. In its simplest form, it requires that if two different groups of researchers carry out an experiment using the same methodologies but *different* samples of the *same* population, it better be the case that the two outcomes of their studies are *statistically indistinguishable.* We investigate this notion in the context of ML (cf. Definition 2.4.6), and characterize for which learning problems statistically indistinguishable learning algorithms exist. Furthermore, we show how statistical indistinguishability, as a property of learning algorithms, is naturally related to various notions of algorithmic stability such as replicability of experiments, and differential privacy.

While we mainly focus on the fundamental ML task of binary classification to make the presentation easier to follow, many of our results extend to other statistical tasks (cf. Section 7.7.2). More formally, the objects of interest are *randomized* learning rules $A : (\mathcal{X} \times \{0,1\})^n \to \{0,1\}^{\mathcal{X}}$. These learning rules take as input a sequence S of n pairs from $\mathcal{X} \times \{0,1\}$, i.e., points from a domain $\mathcal{X}$ along with their labels, and map them to a binary classifier in a randomized manner. We assume that this sequence S is generated i.i.d. from a distribution $\mathcal{D}$ on $\mathcal{X} \times \{0,1\}$. We denote by $\{0,1\}^{\mathcal{X}}$ the space of binary classifiers and by $A(S)$ the random variable that corresponds to the output of A on input S[6]. We also adopt a more algorithmic viewpoint for A where we denote it as a *deterministic* mapping $(\mathcal{X} \times \{0,1\})^n \times \mathcal{R} \to \{0,1\}^{\mathcal{X}}$, which takes as input a training set S of size n made of instance-label pairs and a random string $r \sim \mathcal{R}$ (we use $\mathcal{R}$ for both the probability space and the distribution) corresponding to the algorithm's *internal randomness*, and outputs a hypothesis $A(S,r) \in \{0,1\}^{\mathcal{X}}$. Thus, $A(S)$ corresponds to a random variable while $A(S,r)$ is a deterministic object. To make the distinction clear, we refer to $A(S)$ as (the image of) a *learning rule* and to $A(S,r)$ as (the image of) a *learning algorithm*.

Statistical Indistinguishability

We measure how much two distributions over hypotheses differ using some notion of **statistical dissimilarity** d, which can belong to a quite general class; we could let it be either an Integral Probability Metric (IPM) (e.g., TV or Wasserstein distance, see Definition 7.7.2) or an f-divergence (e.g., KL or Rényi divergence). For further details, see (SFG+09). We are now ready to introduce the following general definition of *indistinguishability of learning rules.*

Definition 2.4.6 (Indistinguishability)**.** *Let d be a statistical dissimilarity measure. A learning rule A is n-sample ρ-indistinguishable with respect to d if for any distribution $\mathcal{D}$ over inputs and two independent sets $S, S' \sim \mathcal{D}^n$ it holds that*

$$\mathop{\mathbb{E}}_{S,S' \sim \mathcal{D}^n} \left[d\left(A(S), A(S')\right)\right] \leq \rho\,.$$

In words, Definition 2.4.6 states that the expected dissimilarity of the outputs of the learning rule when executed on two training sets that are drawn independently from $\mathcal{D}$ is small. We view Definition 2.4.6 as a general information-theoretic way to study indistinguishability as a property of learning rules. In particular, it captures the property that the distribution of outcomes of a learning rule being *indistinguishable* under the resampling of its inputs. Definition 2.4.6 provides the flexibility to define the dissimilarity measure according to the needs of the application domain. For instance, it captures as a special case the global stability property (BLM20) (see Section 7.7.2).

Replicability

Since the issue of replicability is omnipresent in scientific disciplines it is important to design a formal framework through which we can argue about the replicability of experiments. Recently, various works proposed algorithmic definitions of replicability in the context of learning from samples (ILPS22; BGH+23), optimization (AJJ+22), bandits (EKK+22) and clustering (EKM+23), and designed algorithms

[6] We identify with $A(S)$ the posterior distribution of A on input S when there is no confusion.

that are provably replicable under these definitions. A notion that is closely related to Definition 2.4.6 was introduced by (ILPS22): reproducibility or replicability[7] of learning algorithms is defined as follows:

Definition 2.4.7 (Replicability (ILPS22))**.** *Let $\mathcal{R}$ be a distribution over random strings. A learning algorithm A is n-sample ρ-replicable if for any distribution $\mathcal{D}$ over inputs and two independent sets $S, S' \sim \mathcal{D}^n$ it holds that*

$$\Pr_{S,S' \sim \mathcal{D}^n, r \sim \mathcal{R}}[A(S,r) \neq A(S',r)] \leq \rho\,.$$

The existence of a shared random seed r in the definition of replicability is one of the main distinctions between Definition 2.4.6 and 2.4.7. This shared random string can be seen as a way to achieve a *coupling* (see Definition 7.7.1) between two executions of the algorithm A. An interesting aspect of this definition is that replicability is verifiable; replicability under Definition 2.4.7 can be tested using polynomially many samples, random seeds r and queries to A. We remark that the work of (GKM21) introduced the closely related notion of pseudo-global stability (see Definition 7.3.2); the definitions of replicability and pseudo-global stability are equivalent up to polynomial factors in the parameters.

Differential Privacy

The notions of algorithmic indistinguishability and replicability that we have discussed so far have close connections with the classical definition of approximate differential privacy (DR14). For $a, b, \epsilon, \delta \in [0,1]$, let $a \approx_{\epsilon,\delta} b$ denote the statement $a \leq e^{\epsilon} b + \delta$ and $b \leq e^{\epsilon} a + \delta$. We say that two probability distributions P, Q are (ϵ, δ)-indistinguishable if $P(E) \approx_{\epsilon,\delta} Q(E)$ for any measurable event E.

Definition 2.4.8 (Approximate Differential Privacy (DKM+06))**.** *A learning rule A is an n-sample (ϵ, δ)-differentially private if for any pair of samples $S, S' \in (\mathcal{X} \times \{0,1\})^n$ that disagree on a single example, the induced posterior distributions $A(S)$ and $A(S')$ are (ϵ, δ)-indistinguishable.*

We remind the reader that, in the context of PAC learning, any hypothesis class $\mathcal{H}$ can be PAC-learned by an approximate differentially-private algorithm if and only if it has a finite Littlestone dimension $\mathrm{Ldim}(\mathcal{H})$ (see Definition 7.7.3), i.e., there is a qualitative equivalence between online learnability and private PAC learnability (ALMM19; BLM20; GGKM21; ABL+22).

Having defined the standard stability notions required for this chapter, we are now ready to introduce Total Variation Indistinguishability, which is a special instantiation of statistical indistinguishability.

[7] This property was originally defined as "reproducibility" in (ILPS22), but later it was pointed out that the correct term for this definition is "replicability" (see also (BGH+23)). We use the term replicability throughout our work.

TV Indistinguishable Learning Rules

As we discussed, our Definition 2.4.6 captures the property of a learning rule having *indistinguishable* outcomes under the resampling of its inputs from the same distribution. In what follows, we instantiate Definition 2.4.6 with d being the total variation (TV) distance, probably the most well-studied notion of statistical distance in theoretical computer science. Total variation distance between two distributions P and Q over the probability space (Ω, Σ_Ω) can be expressed as

$$\begin{aligned} d_{\mathrm{TV}}(P,Q) &= \sup_{A \in \Sigma_\Omega} P(A) - Q(A) \\ &= \inf_{(X,Y)\sim\Pi(P,Q)} \mathbf{Pr}[X \neq Y]\,, \end{aligned} \tag{2.1}$$

where the infimum is over all couplings between P and Q so that the associated marginals are P and Q respectively. A *coupling* between the distributions P and Q is a set of variables (X, Y) on some common probability space with the given marginals, i.e., $X \sim P$ and $Y \sim Q$. We think of a coupling as a construction of random variables X, Y with prescribed laws.

Setting $d = d_{\mathrm{TV}}$ in Definition 2.4.6, we get the following natural definition. For simplicity, we use the term TV indistinguishability to capture indistinguishability with respect to the TV distance.

Definition 2.4.9 (Total Variation Indistinguishability)**.** *A learning rule A is n-sample ρ-*TV *indistinguishable if for any distribution over inputs $\mathcal{D}$ and two independent sets $S, S' \sim \mathcal{D}^n$ it holds that*

$$\mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S), A(S'))] \leq \rho\,.$$

For some equivalent definitions, we refer to Section 7.7.3. Moreover, for some extensive discussion about the motivation of this definition, see Section 7.7.5. We emphasize that the notion of TV distance has very strong connections with statistical indistinguishability of distributions. If two distributions P and Q are close in TV distance, then, intuitively, no statistical test can distinguish whether an observation was drawn from P or Q. In particular, if $d_{\mathrm{TV}}(P,Q) = \rho$, then $\rho/2$ is the maximum advantage an analyst can achieve in determining whether a random sample X came from P or from Q (where P or Q is used with probability $1/2$ each). In what follows, we focus on this particular notion of statistical dissimilarity.

Our Contribution on Challenge 5. In this chapter, we investigate the connections between TV indistinguishability, replicability and differential privacy.

TV Indistinguishability $\iff$ Replicability. We show that TV indistinguishability and replicability are equivalent. This equivalence holds for countable domains[8] and extends to general statistical tasks (cf. Section 7.9.2).

[8] We remark that the direction replicability implies TV indistinguishability holds for general domains.

Informal Theorem. *The following hold true.*

- *If a learning rule A is n-sample ρ-replicable, then it is also n-sample ρ-*TV *indistinguishable.*
- *Let $\mathcal{X}$ be a countable domain and let A be a learning rule that is n-sample ρ-*TV *indistinguishable. Then, there exists an equivalent learning rule A' that is n-sample $\frac{2\rho}{1+\rho}$-replicable.*

We remark that our transformations between replicable and TV indistinguishable learners do not change the (possibly randomized) input → output map which is induced by the learner; i.e., given a TV indistinguhishable learner $\mathcal{A}$, we transform it to a replicable learner $\mathcal{A}'$ such that $\mathcal{A}(S)$ and $\mathcal{A}'(S)$ are the same distributions over output hypotheses for every input sample S.

At this point we would like to highlight a subtle difference between replicability and other well studied notions of stability that arise in learning theory such as differential privacy, TV indistinguishability, one-way perfect generalization, and others. The latter notions of stability depend only on the input → output map which is induced by the learner. In contrast, the definition of replicability has to do with the way the algorithm is implemented (in particular the way randomness is used). In other words, the definition of replicability enables having two learning rules $\mathcal{A}', \mathcal{A}''$ that compute exactly the same input → output map, but such that $\mathcal{A}'$ is replicable and $\mathcal{A}''$ is not. Thus, our equivalence suggests an interpretation of TV indistinguishability as an abstraction/extension of replicability that only depends on the input-output mechanism.

TV Indistinguishability $\iff$ Approximate DP. We show that TV indistinguishability and (ϵ, δ)-DP are statistically equivalent. This equivalence holds for countable[9] domains in the context of PAC learning. As an intermediate result, we also show that replicability and (ϵ, δ)-DP are statistically equivalent in the context of PAC learning, and this holds for general domains.

Informal Theorem. *The following hold true.*

- *Let $\gamma \in (0, 1/2), \alpha, \beta, \rho \in (0, 1)^3$. Assume that $\mathcal{H}$ is learnable by an n-sample $(1/2 - \gamma, 1/2 - \gamma)$-accurate $(0.1, 1/(n^2 \log(n)))$-differentially private learner. Then, it is also learnable by an (α, β)-accurate ρ-*TV *indistinguishable learning rule.*

[9] We remark that the direction (ϵ, δ)-DP implies TV indistinguishability holds for general domains.

- *Let $\mathcal{X}$ be a countable domain. Assume that $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is learnable by an (α, β)-accurate ρ-TV indistinguishable learner A, for some $\rho \in (0,1), \alpha \in (0,1/2), \beta \in \left(0, \frac{1-\rho}{1+\rho}\right)$. Then, for any $(\alpha', \beta', \varepsilon, \delta) \in (0,1)^4$, it is also learnable by an $(\alpha + \alpha', \beta')$-accurate (ε, δ)-differentially private learner A'.*

We shortly mention that the indepent work of (BGH$^+$23) gives an alternative proof of the equivalence between TV indistinguishability, replicability, and differential privacy. In contrast with our equivalence, the transformations by (BGH$^+$23) are restricted to finite classes. On the other hand, (BGH$^+$23) give a constructive proof whereas our proof is purely information-theoretic. For further discussion, we refer to Chapter 7.

Boosting and Amplification. Finally, we provide statistical amplification and boosting algorithms for TV indistinguishable learners for countable domains.

Informal Theorem. *Let $\alpha, \beta, \rho \in (0,1)^2$ denote the accuracy, confidence and* TV *indistinguishability parameters respectively. For every countable domain $\mathcal{X}$, any concept class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, any non-trivial TV indistinguishable learner for $\mathcal{H}$ can be boosted so that $\alpha, \beta, \rho \to 0$.*

En route, we improve the sample complexity of some routines provided in (ILPS22). These results can be found in Chapter 7.

2.5 Bibliographic Information

The results described in this thesis have already appeared in existing publications, which we briefly mention below.

Chapter 3 is based on (FKT20) that appeared in COLT 2020. Chapter 4 is based on (FKKT21) that appeared in COLT 2021. Chapter 5 is based on (FKKT22) that appeared in NeurIPS 2022 as an Oral. Chapter 6 is based on (EKM$^+$23) that appeared in ICLR 2023. Chapter 7 is based on (KKMV23) that appeared in ICML 2023.

Other papers by the author over the course of his PhD that are not included in this thesis are (FKS21; FKP21; KSZ22; MKFI22; FKT22; KVK22).

Chapter 3

Learning from Truncated Samples

In this chapter, we are going to investigate efficient parameter estimation of discrete models from truncated samples. We consider the fundamental setting of a Boolean product distribution $\mathcal{D}$ on the d-dimensional hypercube truncated by a set S, which is accessible through membership queries. The marginal of $\mathcal{D}$ in each direction i is an independent Bernoulli distribution with parameter $p_i \in (0, 1)$. Our goal is to compute an estimation $\widehat{\boldsymbol{p}}$ of the parameter vector $\boldsymbol{p}$ of $\mathcal{D}$ such that $\|\boldsymbol{p} - \widehat{\boldsymbol{p}}\|_2 \leq \epsilon$, with probability of at least $1 - \delta$, with time and sample complexity polynomial in d, $1/\epsilon$ and $\log(1/\delta)$. We note that such an estimation $\widehat{\boldsymbol{p}}$ (or an estimation $\widehat{\boldsymbol{z}}$ of the logit parameters $\boldsymbol{z} = (\log \frac{p_1}{1-p_1}, \ldots, \log \frac{p_d}{1-p_d})$ of similar accuracy) implies an estimation of the true distribution within total variation distance ϵ.

We develop novel techniques for truncated statistics for discrete distributions. As aforementioned, there has been a large number of recent works dealing inference with truncated data from a Gaussian distribution (DGTZ18; KTZ19; DGTZ19a) or mixtures of Gaussians (NP19) but to the best of our knowledge there is no work dealing with discrete distributions. An additional feature of our work compared to those results is that our methods are not limited to parameter estimation but enable any statistical task to be performed on truncated datasets by providing a sampler to the true underlying distribution. While this requires a mildly stronger than necessary but natural assumption on the truncation set, we show that the more complex SGD based methods developed in prior work can also be applied in the discrete settings we consider.

The field of robust statistics is also very related to our work as it also deals with biased data-sets and aims to identify the distribution that generated the data. Truncation can be seen as an adversary erasing samples outside a certain set. Recently, there has been a lot of theoretical work for computationally-efficient robust estimation of high-dimensional distributions in the presence of arbitrary corruptions to a small ε fraction of the samples, allowing for both deletions of samples and additions of samples (DKK$^+$16; CSV17; LRV16b; DKK$^+$17; DKK$^+$18; HL19). In particular, the work of (DKK$^+$16) deals with the problem of learning binary-product distributions.

Another line of related work concerns learning from positive examples. The work of (DDS14) considers a setting where samples are obtained from the uniform distribution over the hypercube truncated on a set S. However, their goal is somewhat orthogonal to ours. It aims to accurately learn the set S while the distribution is already known. In contrast, in our setting the truncation set is known and the goal is to learn the distribution. More recently, (CDS20) extend these results to learning the truncation set with truncated samples from continuous distributions.

Another related literature within learning theory aims to learn discrete distributions through conditional samples. In the conditional sampling model that was recently introduced concurrently by (CFGM13) and (CRS14; CRS15), the goal is again to learn an underlying discrete distribution through conditional/truncated samples but the learner can change the truncation set on demand. This is known to be a more powerful model for distribution learning and testing than standard sampling (Can15; FJO$^+$15; ACK15b; BC18; ACK15a; GTZ17; KT19; CCK$^+$19).

3.1 Preliminaries

We use lowercase bold letters $\boldsymbol{x}$ to denote d-dimensional vectors. We let $\|\boldsymbol{x}\|_p = (\sum_{i=1}^d |x_i|^p)^{1/p}$ denote the L_p norm and $\|\boldsymbol{x}\|_\infty = \max_{i\in[d]}\{|x_i|\}$ denote the L_∞ norm of a vector $\boldsymbol{x}$. We let $[d] \overset{\text{def}}{=} \{1,\dots,d\}$ and $\Pi_d = \{0,1\}^d$ denotes the d-dimensional Boolean hypercube.

For any vector $\boldsymbol{x}$, $\boldsymbol{x}_{-i}$ is the vector obtained from $\boldsymbol{x}$ by removing the i-th coordinate and $(\boldsymbol{x}_{-i}, y)$ is the vector obtained from $\boldsymbol{x}$ by replacing x_i by y. Similarly, given a set $S \subseteq \Pi_d$, we let $S_{-i} = \{\boldsymbol{x}_{-i} : (\boldsymbol{x}_{-i}, 0) \in S \vee (\boldsymbol{x}_{-i}, 1) \in S\}$ be the projection of S to $\Pi_{[d]\setminus\{i\}}$. For any $\boldsymbol{x} \in \Pi_d$ and any coordinate $i \in [d]$, we let $\textsc{flip}(\boldsymbol{x}, i) = (\boldsymbol{x}_{-i}, 1 - x_i)$ denote $\boldsymbol{x}$ with its i-th coordinated flipped.

Bernoulli Distribution. For any $p \in [0,1]$, we let $\mathcal{B}e(p)$ denote the Bernoulli distribution with parameter p. For any $x \in \{0,1\}$, $\mathcal{B}e(p; x) = p^x(1-p)^{1-x}$ denotes the probability of value x under $\mathcal{B}e(p)$. The Bernoulli distribution is an exponential family[1], where the natural parameter, denoted z, is the logit $z = \log \frac{p}{1-p}$ of the parameter p[2]. The inverse parameter mapping is $p = \frac{1}{1+\exp(-z)}$. Also, the base measure is $h(x) = 1$, the sufficient statistic is the identity mapping $T(x) = x$ and the log-partition function with respect to p is $\alpha(p) = -\log(1-p)$.

Boolean Product Distribution. We mostly focus on the fundamental family of *Boolean product distributions* on the d-dimensional hypercube Π_d. A Boolean

[1] The exponential family $\mathcal{E}(\boldsymbol{T}, h)$ with sufficient statistics $\boldsymbol{T}$, carrier measure h and natural parameters $\boldsymbol{\eta}$ is the family of distributions $\mathcal{E}(\boldsymbol{T}, h) = \{\mathcal{P}_{\boldsymbol{\eta}} : \boldsymbol{\eta} \in \mathcal{H}_{\boldsymbol{T},h}\}$, where the probability distribution $\mathcal{P}_{\boldsymbol{\eta}}$ has density $p_{\boldsymbol{\eta}}(x) = h(x)\exp(\boldsymbol{\eta}^T\boldsymbol{T}(x) - \alpha(\boldsymbol{\eta}))$, where α is the log-partition function.

[2] The base of the logarithm function log used throughout the paper is insignificant.

product distribution with parameter vector $\boldsymbol{p} = (p_1, \ldots, p_d)$, usually denoted by $\mathcal{D}(\boldsymbol{p})$, is the product of d independent Bernoulli distributions, i.e., $\mathcal{D}(\boldsymbol{p}) = \mathcal{B}e(p_1) \otimes \cdots \otimes \mathcal{B}e(p_d)$. The Boolean product distribution can be expressed in the form of an exponential family as follows:

$$\mathcal{D}(\boldsymbol{z}; \boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^T \boldsymbol{z})}{\prod_{i \in [d]}(1 + \exp(z_i))}\,, \tag{3.1}$$

where $\boldsymbol{z} = (z_1, \ldots, z_d)$ is the natural parameter vector with $z_i = \log \frac{p_i}{1-p_i}$ for each $i \in [d]$.

In the following, we always let $\mathcal{D}$ (or $\mathcal{D}(\boldsymbol{p})$ or $\mathcal{D}(\boldsymbol{z})$, when we want to emphasize the parameter vector $\boldsymbol{p}$ or the natural parameter vector $\boldsymbol{z}$) denote a Boolean product distribution. We denote $\boldsymbol{z}(\boldsymbol{p})$ (or simply $\boldsymbol{z}$, when $\boldsymbol{p}$ is clear from the context) the vector of natural parameters of $\mathcal{D}$. We let $\mathcal{D}(\boldsymbol{p}; \boldsymbol{x})$ and $\mathcal{D}(\boldsymbol{z}; \boldsymbol{x})$ (or simply $\mathcal{D}(\boldsymbol{x})$, when $\boldsymbol{p}$ or $\boldsymbol{z}$ are clear from the context) denote the probability of $\boldsymbol{x} \in \Pi_d$ under $\mathcal{D}$. Given a subset $S \subseteq \Pi_d$ of the Boolean hypercube, the probability mass assigned to S by a distribution $\mathcal{D}(\boldsymbol{p})$, usually denoted $\mathcal{D}(\boldsymbol{p}; S)$ (or simply $\mathcal{D}(S)$, when $\boldsymbol{p}$ is clear from the context), $\mathcal{D}(\boldsymbol{p}; S) = \sum_{\boldsymbol{x} \in S} \mathcal{D}(\boldsymbol{p}; \boldsymbol{x})$.

Truncated Boolean Product Distribution. Given a Boolean product distribution $\mathcal{D}$, we define the *truncated Boolean product distribution* $\mathcal{D}_S$, for any fixed $S \subseteq \Pi_d$. $\mathcal{D}_S$ has $\mathcal{D}_S(\boldsymbol{x}) = \mathcal{D}(\boldsymbol{x})/\mathcal{D}(S)$, for all $\boldsymbol{x} \in S$, and $\mathcal{D}_S(\boldsymbol{x}) = 0$, otherwise. We often refer to $\mathcal{D}_S$ as the truncation of $\mathcal{D}$ (by S) and to S as the *truncation set.*

It is sometimes convenient (especially when we discuss assumptions 1 and 3, in Section 3.3), to refer to some fixed element of S. We observe that by swapping 1 with 0 (and p_i with $1 - p_i$) in certain directions, we can *normalize* S so that $\mathbf{0} \in S$ and $\mathcal{D}_S(\mathbf{0}) > 0$. In the following, we always assume, without loss of generality, that S is normalized so that $\mathbf{0} \in S$ and $\mathcal{D}_S(\mathbf{0}) > 0$.

Notions of Distance between Distributions. Let $\mathrm{P}, \mathcal{Q}$ be two probability measures in the discrete probability space $(\Omega, \mathcal{F})$. The *total variation distance* between P and $\mathcal{Q}$, denoted $d_{\mathrm{TV}}(\mathrm{P}, \mathcal{Q})$, is defined as $d_{\mathrm{TV}}(\mathrm{P}, \mathcal{Q}) = \frac{1}{2}\sum_{x \in \Omega} |\mathrm{P}(x) - \mathcal{Q}(x)| = \max_{A \in \mathcal{F}} |\mathrm{P}(A) - \mathcal{Q}(A)|$. The *Kullback–Leibler divergence* (or simply, *KL divergence*), denoted $D_{KL}(\mathrm{P} \parallel \mathcal{Q})$, is defined as $D_{KL}(\mathrm{P} \parallel \mathcal{Q}) = \mathbb{E}_{x \sim \mathrm{P}}\left[\log \frac{\mathrm{P}(x)}{\mathcal{Q}(x)}\right] = \sum_{x \in \Omega} \mathrm{P}(x) \log \frac{\mathrm{P}(x)}{\mathcal{Q}(x)}$. We first recall that the KL divergence is additive for product distributions.

Proposition 3.1.1. *Let $\mathcal{P}(\boldsymbol{p})$ and $\mathcal{Q}(\boldsymbol{q})$ be two Boolean product distributions. Then,*

$$D_{KL}(\mathcal{P} \parallel \mathcal{Q}) = \sum_{i=1}^{d} \left(p_i \log \frac{p_i}{q_i} + (1 - p_i) \log \frac{1 - p_i}{1 - q_i} \right)\,. \tag{3.2}$$

Next, we observe that for two Bernoulli distributions, with parameters p and q, the KL divergence can be upper bounded by the squared distance of their natural parameters. We provide the proof of Proposition 3.1.2 in the Section 3.6.

Proposition 3.1.2. *For all $p, q \in (0,1)$, the following holds:*

$$D_{KL}\big(\mathcal{B}e(p) \parallel \mathcal{B}e(q)\big) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q} \leq \left(\log \frac{p}{1-p} - \log \frac{q}{1-q} \right)^2 .$$

The following summarizes some standard upper bounds on the total variation distance and the KL divergence of two Boolean product distributions.

Proposition 3.1.3. *Let $\mathcal{P}(\boldsymbol{p})$ and $\mathcal{Q}(\boldsymbol{q})$ be two Boolean product distributions with $\boldsymbol{p}, \boldsymbol{q} \in (0,1)^d$, and let $\boldsymbol{z}(\boldsymbol{p})$ and $\boldsymbol{z}(\boldsymbol{q})$ be the vectors of their natural parameters. Then, the following hold:*

(i) $D_{KL}(\mathcal{P} \parallel \mathcal{Q}) \leq \|\boldsymbol{z}(\boldsymbol{p}) - \boldsymbol{z}(\boldsymbol{q})\|_2^2$.

(ii) $d_{\mathrm{TV}}(\mathcal{P}, \mathcal{Q}) \leq \frac{\sqrt{2}}{2} \|\boldsymbol{z}(\boldsymbol{p}) - \boldsymbol{z}(\boldsymbol{q})\|_2$.

(iii) $d_{\mathrm{TV}}(\mathcal{P}, \mathcal{Q}) \leq \sqrt{2 \cdot \sum_{i=1}^d \frac{(p_i - q_i)^2}{(p_i + q_i)(2 - p_i - q_i)}}$.

Now Proposition 3.1.3 is an immediate consequence of Proposition 3.1.1, Proposition 3.1.2 and Pinsker's inequality (for (i) and (ii)), and (DKK$^+$16, Lemma 2.17) (for (iii)).

Identifiability and Learnability. A Boolean product distribution $\mathcal{D}(\boldsymbol{p})$ is *identifiable* from its truncation $\mathcal{D}_S(\boldsymbol{p})$, if given $\mathcal{D}_S(\boldsymbol{p}; \boldsymbol{x})$, for all $\boldsymbol{x} \in S$, we can recover the parameter vector $\boldsymbol{p}$.

A Boolean product distribution $\mathcal{D}(\boldsymbol{p})$ is *efficiently learnable* from its truncation $\mathcal{D}_S(\boldsymbol{p})$, if for any $\epsilon, \delta > 0$, we can compute an estimation $\widehat{\boldsymbol{p}}$ of the parameter vector $\boldsymbol{p}$ (or an estimation $\widehat{\boldsymbol{z}}$ of the natural parameter vector $\boldsymbol{z}$) of $\mathcal{D}$ such that $\|\boldsymbol{p} - \widehat{\boldsymbol{p}}\|_2 \leq \epsilon$ (or $\|\boldsymbol{z} - \widehat{\boldsymbol{z}}\|_2 \leq \epsilon$), with probability at least $1 - \delta$, with time and sample complexity polynomial in d, $1/\epsilon$ and $\log(1/\delta)$ using truncated samples from $\mathcal{D}_S(\boldsymbol{p})$. By Proposition 3.1.3, an upper bound on the L_2 distance between $\widehat{\boldsymbol{z}}$ and $\boldsymbol{z}$ (or between $\widehat{\boldsymbol{p}}$ and $\boldsymbol{p}$) translates into an upper bound on the total variation distance between the true distribution and $\mathcal{D}(\widehat{\boldsymbol{z}})$ (or $\mathcal{D}(\widehat{\boldsymbol{p}})$). In this chapter, we identify sufficient and necessary conditions for efficient learnability of Boolean product distributions from truncated samples.

3.2 Boolean Product Distributions Truncated by Fat Sets

In this section, we discuss *fatness* of the truncation set, a strong sufficient (and in a certain sense, necessary) condition, under which we can generate samples from a Boolean product distribution $\mathcal{D}$ using samples from its truncation $\mathcal{D}_S$ (and access to S through a membership oracle).

Definition 3.2.1. *A truncated Boolean product distribution $\mathcal{D}_S$ is α-fat in coordinate $i \in [d]$, for some $\alpha > 0$, if $\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{D}_S}[\textsc{flip}(\boldsymbol{x}, i) \in S] \geq \alpha$. A truncated Boolean product distribution $\mathcal{D}_S$ is α-fat, for some $\alpha > 0$, if $\mathcal{D}_S$ is α-fat in every coordinate $i \in [d]$.*

If $\mathcal{D}_S$ is fat, it happens often that a sample $\boldsymbol{x} \sim \mathcal{D}_S$ has both $(\boldsymbol{x}_{-i}, 0), (\boldsymbol{x}_{-i}, 1) \in S$. Then, conditional on the remaining coordinates $\boldsymbol{x}_{-i}$, the i-th coordinate x_i of $\boldsymbol{x}$ is distributed as $\mathcal{B}e(p_i)$. We next focus on truncated Boolean product distributions $\mathcal{D}_S$ that are α-fat.

There are several natural classes of truncation subsets that give rise to fat truncated product distributions. E.g., for each $k \in [d]$, the halfspace $S_{\leq k} = \{\boldsymbol{x} \in \Pi_d : x_1 + \ldots + x_d \leq k\}$ results in an α-fat truncated distribution, if $\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{D}_{S_{\leq k}}}[x_i = 1] \geq \alpha$, for all $i \in [d]$. The same holds if S is any *downward closed*[3] subset of Π_d and $\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{D}_S}[x_i = 1] \geq \alpha$, for all $i \in [d]$.

Fatness in coordinate $i \in [d]$ is necessary, if we want to distinguish between two truncated Boolean distributions based on their i-th parameter only, if the remaining coordinates are correlated. Specifically, we can show that if $\mathcal{D}_S$ is 0-fat in some coordinate i, there exists a Boolean distribution with $q_i \neq p_i$ (and $|q_i - p_i|$ large enough) whose truncation by S appears identical to $\mathcal{D}_S$. Therefore, if the other coordinates are arbitrarily correlated, it is impossible to distinguish between the two distributions based on their i-th parameter alone. However, as we discuss in Section 3.3, if S is rich enough, but not necessarily fat, we can recover the entire parameter vector[4] of $\mathcal{D}$.

Lemma 3.2.2. *Let $i \in [d]$, let S be any subset of Π_d with $\textsc{flip}(\boldsymbol{x}, i) \notin S$, for all $\boldsymbol{x} \in S$, and consider any $0 < p < q < 1$. Then, for any Boolean distribution $\mathcal{D}_{-i}$ with $\mathcal{D}_{-i}(S_{-i}) \in (0,1)$, there exists a distribution $\mathcal{D}'_{-i}$ such that $(\mathcal{B}e(p) \otimes \mathcal{D}_{-i})_S \equiv (\mathcal{B}e(q) \otimes \mathcal{D}'_{-i})_S$.*

Proof. We recall that $S_{-i} = \{\boldsymbol{x}_{-i} : (\boldsymbol{x}_{-i}, 0) \in S \vee (\boldsymbol{x}_{-i}, 1) \in S\}$ denotes the projection of S on $\Pi_{[d]\setminus\{i\}}$. By hypothesis, $|S| = |S_{-i}|$ and for each $\boldsymbol{x}_{-i} \in S_{-i}$, either $(\boldsymbol{x}_{-i}, 0) \in S$ or $(\boldsymbol{x}_{-i}, 1) \in S$, but never both. For each $\boldsymbol{x}_{-i} \in S_{-i}$, we let:

$$\mathcal{D}'_{-i}(\boldsymbol{x}_{-i}) = \begin{cases} \mathcal{D}_{-i}(\boldsymbol{x}_{-i}) \frac{p}{q} & \text{if } (\boldsymbol{x}_{-i}, 1) \in S\,, \\ \mathcal{D}_{-i}(\boldsymbol{x}_{-i}) \frac{1-p}{1-q} & \text{if } (\boldsymbol{x}_{-i}, 0) \in S\,. \end{cases}$$

[3] A set $S \subseteq \Pi_d$ is downward closed if for any $\boldsymbol{x} \in S$ and any $\boldsymbol{y}$ with $y_i \leq x_i$, in all directions $i \in [d]$, $\boldsymbol{y} \in S$.

[4] For a concrete example, where we can recover the entire parameter vector of a truncated Boolean product distribution $\mathcal{D}_S$, we consider $S = \{000, 110, 011, 101\} \subseteq \Pi_3$, which is not fat in any coordinate, and let $p_{\boldsymbol{x}} = \mathcal{D}_S(\boldsymbol{x})$, for each $\boldsymbol{x} \in S$. Then, setting $z_i = \log \frac{p_i}{1-p_i}$, for each i, we can recover (p_1, p_2, p_3), by solving the following linear system: $z_1 + z_2 = \log \frac{p_{110}}{p_{000}}$, $z_2 + z_3 = \log \frac{p_{011}}{p_{000}}$, $z_1 + z_3 = \log \frac{p_{101}}{p_{000}}$. This is a special case of the more general identifiability condition discussed in Lemma 3.3.1.

For each $\boldsymbol{y} \in \Pi_{d-1} \setminus S_{-i}$, we let $\mathcal{D}'_{-i}(\boldsymbol{y}) \propto \mathcal{D}_{-i}(\boldsymbol{y})$, so that $\mathcal{D}'_{-i}$ is a probability distribution on Π_{d-1} . E.g., if for all $\boldsymbol{x}_{-i} \in S_{-i}$, $(\boldsymbol{x}_{-i}, 1) \in S$, we let

$$\mathcal{D}'_{-i}(\boldsymbol{y}) = \mathcal{D}_{-i}(\boldsymbol{y}) \frac{1 - \mathcal{D}_{-i}(S_{-i})\frac{p}{q}}{1 - \mathcal{D}_{-i}(S_{-i})} .$$

By definition, $\mathcal{B}e(q) \otimes \mathcal{D}'_{-i}$ is a probability distribution on Π_d. Moreover, for all $\boldsymbol{x} \in S$, $(\mathcal{B}e(p) \otimes \mathcal{D}_{-i})(\boldsymbol{x}) = (\mathcal{B}e(q) \otimes \mathcal{D}'_{-i})(\boldsymbol{x})$, which implies the lemma.

□

3.2.1 Sampling from a Boolean Product Distribution using Samples from its Fat Truncation

An interesting consequence of fatness is that we can efficiently generate samples from a Boolean product distribution $\mathcal{D}$ using samples from any α-fat truncation of $\mathcal{D}$. The idea is described in Algorithm 1. Theorem 3.2.3 shows that for any sample $\boldsymbol{x}$ drawn from $\mathcal{D}_S$ and any $i \in [d]$ such that $\textsc{flip}(\boldsymbol{x}, i) \in S$, conditional on $\boldsymbol{x}_{-i}$, x_i is distributed as $\mathcal{B}e(p_i)$. So, we can generate a random sample $\boldsymbol{y} \sim \mathcal{D}$ by putting together d such values. α-fatness of the truncated distribution $\mathcal{D}_S$ implies that the expected number of samples $\boldsymbol{x} \sim \mathcal{D}_S$ required to generate a $\boldsymbol{y} \sim \mathcal{D}$ is $O(\log(d)/\alpha)$.

Algorithm 1 Sampling from $\mathcal{D}$ using samples from $\mathcal{D}_S$

```
procedure SAMPLER($\mathcal{D}_S$)                          ▷ $\mathcal{D}_S$ is $\alpha$-fat.
    $\boldsymbol{y} \leftarrow (-1, \dots -1)$
    while $\exists y_i = -1$ do
        Draw sample $\boldsymbol{x} \sim \mathcal{D}_S$
        for $i \leftarrow 1, \dots, d$ do
            if $\textsc{flip}(\boldsymbol{x}, i) \in S$ then        ▷ We assume oracle access to $S$.
                $y_i \leftarrow x_i$
    return $\boldsymbol{y}$
```

Theorem 3.2.3. *Let $\mathcal{D}$ be a Boolean product distribution over Π_d and let $\mathcal{D}_S$ be any α-fat truncation of $\mathcal{D}$. Then, (i) the distribution of the samples generated by Algorithm 1 is identical to $\mathcal{D}$; and (ii) the expected number of samples from $\mathcal{D}_S$ before a sample is returned by Algorithm 1 is $O(\log(d)/\alpha)$.*

Proof. Let $\widetilde{\mathcal{D}}$ be the distribution of the samples generated by Algorithm 1. To prove that $\mathcal{D}$ and $\widetilde{\mathcal{D}}$ are identical, we show that $\widetilde{D}$ is a product distribution and that each $y_i \sim \mathcal{B}e(p_i)$, where p_i is the parameter of $\mathcal{D}$ in direction $i \in [d]$.

We fix a direction $i \in [d]$. Let $\mathcal{D}_{-i}$ denote the projection of $\mathcal{D}$ on $\Pi_{[d]\setminus\{i\}}$. In Algorithm 1, y_i takes the value of the i-coordinate of a sample $\boldsymbol{x} \sim \mathcal{D}_S$ such that both $(\boldsymbol{x}_{-i}, 0), (\boldsymbol{x}_{-i}, 1) \in S$. For each such sample $\boldsymbol{x}$, we have that:

$$\mathcal{D}_S((\boldsymbol{x}_{-i}, 1)) = \frac{\mathcal{D}_{-i}(\boldsymbol{x}_{-i})\, p_i}{\mathcal{D}(S)} \quad \text{and} \quad \mathcal{D}_S(\boldsymbol{x}_{-i}, 0) = \frac{\mathcal{D}_{-i}(\boldsymbol{x}_{-i})\,(1-p_i)}{\mathcal{D}(S)} . \tag{3.1}$$

Therefore, $\frac{\mathcal{D}_S((\boldsymbol{x}_{-i},1))}{\mathcal{D}_S((\boldsymbol{x}_{-i},0))} = \frac{p_i}{1-p_i}$, which implies that $\mathcal{D}_S((\boldsymbol{x}_{-i}, 1)) = p_i$. Since this holds for all $\boldsymbol{x}_{-i}$ such that both $(\boldsymbol{x}_{-i}, 0), (\boldsymbol{x}_{-i}, 1) \in S$, y_i is independent of the remaining coordinates $\boldsymbol{y}_{-i}$ and is distributed as $\mathcal{B}e(p_i)$. This concludes the proof of *(i)*.

As for the sample complexity of Algorithm 1, we observe that since $\mathcal{D}_S$ is α-fat in each coordinate i, each new sample $\boldsymbol{x}$ covers any fixed coordinate y_i (i.e., $\boldsymbol{x}$ causes y_i to become x_i) of $\boldsymbol{y}$ with probability at least α. Therefore, the probability that any fixed coordinate y_i remains -1 after Algorithm 1 draws k samples from $\mathcal{D}_S$ is at most $(1-\alpha)^k \le e^{-\alpha k}$. Setting $k = 2\log(d)/\alpha$ and applying the union bound, we get that the probability that there is a coordinate of $\boldsymbol{y}$ with value -1 after $2\log(d)/\alpha$ samples from $\mathcal{D}_S$ is at most $de^{-\alpha k} = de^{-2\log(d)} = 1/d$. Therefore, the expected number of samples from $\mathcal{D}_S$ before a random sample $\boldsymbol{y} \sim \mathcal{D}$ is returned by Algorithm 1 is at most

$$\frac{2\log(d)}{\alpha} + \sum_{\ell=0}^{\infty} \frac{e^{-\ell\alpha}}{d} \le \frac{2\log(d)}{\alpha} + \frac{2}{d\alpha} = O\left(\frac{2\log(d)}{\alpha}\right) ,$$

where the inequality follows from $1/(1-e^{-\alpha}) \le 2/\alpha$ for $\alpha \in (0,1)$.

□

3.2.2 Parameter Estimation and Learning in Total Variation Distance

Based on Algorithm 1, we can recover the parameters of any Boolean product distribution $\mathcal{D}$ using samples from any fat truncation of $\mathcal{D}$.

Theorem 3.2.4. *Let $\mathcal{D}(\boldsymbol{p})$ be a Boolean product distribution and let $\mathcal{D}_S(\boldsymbol{p})$ be a truncation of $\mathcal{D}$. If $\mathcal{D}_S$ is α-fat in any fixed coordinate i, then, for any $\epsilon, \delta > 0$, we can compute an estimation $\widehat{p}_i$ of the parameter p_i of $\mathcal{D}$ such that $|p_i - \widehat{p}_i| \le \epsilon$, with probability at least $1-\delta$, using an expected number of $O(\log(1/\delta)/(\epsilon^2\alpha))$ samples from $\mathcal{D}_S$.*

Proof. We modify Algorithm 1 to Algorithm 2, so that it generates random samples $y \in \{0, 1\}$ in coordinate i only. As in Theorem 3.2.3.*(i)*, each y of Algorithm 2 is an independent sample from $\mathcal{B}e(p_i)$. Since the truncated distribution $\mathcal{D}_S$ is α-fat, the expected number of samples from $\mathcal{D}_S$, before y is generated, is $1/\alpha$. We estimate p_i from n samples $y^{(1)}, \dots, y^{(n)}$ of Algorithm 2 using the empirical mean $\widehat{p}_i = \sum_{\ell=1}^{n} y^{(\ell)}/n$. A standard application of the Hoeffding bound[5] shows that if

[5] We use the following Hoeffding bound: Let $X_1, \dots, X_n$ be n independent Bernoulli random variables, let $X = \frac{1}{n}(\sum_{i=1}^{n} X_i)$ and $\mathbb{E}[X] = \frac{1}{n}(\sum_{i=1}^{n} \mathbb{E}[X_i])$. Then, for any $t \ge 0$, $\mathbf{Pr}[|X - \mathbb{E}[X]| \ge t] \le 2e^{-2nt^2}$.

Algorithm 2 Sampling coordinate $i \in [d]$ from $\mathcal{D}$ using samples from $\mathcal{D}_S$

```
procedure SAMPLER($\mathcal{D}_S, i$)        ▷ $\mathcal{D}_S$ is fat in coordinate $i$.
    $y \leftarrow -1$
    while $y = -1$ do
        Draw sample $\boldsymbol{x} \sim \mathcal{D}_S$
        if FLIP($\boldsymbol{x}, i$) $\in S$ then        ▷ We have oracle access to $S$.
            $y \leftarrow x_i$
    return $y$
```

$n = \log(2/\delta)/\epsilon^2$, then $|p_i - \widehat{p}_i| \leq \epsilon$, with probability at least $1-\delta$. Hence, estimating p_i with accuracy ϵ requires an expected number of $O(\log(1/\delta)/(\epsilon^2\alpha))$ samples from $\mathcal{D}_S$.

□

Using $n = \log(2d/\delta)/\epsilon^2$ samples $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n)}$ generated by Algorithm 1, we can estimate all the parameters $\boldsymbol{p}$ of $\mathcal{D}$, by taking $\widehat{p}_i = \sum_{\ell=1}^{n} y_i^{(\ell)}/n$, for each $i \in [d]$. The following is an immediate consequence of Theorems 3.2.3 and 3.2.4.

Corollary 3.2.5. *Let $\mathcal{D}(\boldsymbol{p})$ be a Boolean product distribution and $\mathcal{D}_S(\boldsymbol{p})$ be any α-fat truncation of $\mathcal{D}$. Then, for any $\epsilon, \delta > 0$, we can compute an estimation $\widehat{\mathbf{p}}$ such that $\|\boldsymbol{p} - \widehat{\mathbf{p}}\|_\infty \leq \epsilon$, with probability at least $1 - \delta$, using an expected number of $O(\log(d)\log(d/\delta)/(\epsilon^2\alpha))$ samples from $\mathcal{D}_S$.*

3.2.3 Identity and Closeness Testing with Access to Truncated Samples

Theorem 3.2.3 implies that if we have sample access to an α-fat truncation $\mathcal{D}_S$ of a Boolean product distribution $\mathcal{D}$, we can pretend that we have sample access to the original distribution $\mathcal{D}$, at the expense of an increase in the sample complexity (from $\mathcal{D}_S$) by a factor of $O(\log(d)/\alpha)$. Therefore, we can extend virtually all known hypothesis testing and learning algorithms for Boolean product distributions to fat truncated Boolean product distributions.

For *identity testing* of Boolean product distributions, based on samples from fat truncated ones, we combine Algorithm 1 with the algorithm of (CDKS17, Sec. 4.1). Combining Theorem 3.2.3 with (CDKS17, Theorem 6), we obtain the following:

Corollary 3.2.6 (Identity Testing)**.** *Let $\mathcal{Q}(\boldsymbol{q})$ be a Boolean product distribution described by its parameters $\boldsymbol{q}$, and let $\mathcal{D}$ be a Boolean product distribution for which we have sample access to its α-fat truncation $\mathcal{D}_S$. For any $\epsilon > 0$, we can distinguish between $d_{\mathrm{TV}}(\mathcal{Q}, \mathcal{D}) = 0$ and $d_{\mathrm{TV}}(\mathcal{Q}, \mathcal{D}) > \epsilon$, with probability $2/3$, using an expected number of $O(\log(d)\sqrt{d}/(\alpha\epsilon^2))$ samples from $\mathcal{D}_S$.*

We can extend Corollary 3.2.6 to *closeness testing* of two Boolean product distributions, for which we only have sample access to their fat truncations. We combine Algorithm 1 with the algorithm of (CDKS17, Sec. 5.1). The following is an immediate consequence of Theorem 3.2.3 and (CDKS17, Theorem 9).

Corollary 3.2.7 (Closeness Testing)**.** *Let $\mathcal{Q}$, $\mathcal{D}$ be two Boolean product distributions for which we have sample access to their α_1-fat truncation $\mathcal{Q}_{S_1}$ and α_2-fat truncation $\mathcal{D}_{S_2}$. For any $\epsilon > 0$, we can distinguish between $d_{\mathrm{TV}}(\mathcal{Q},\mathcal{D}) = 0$ and $d_{\mathrm{TV}}(\mathcal{Q},\mathcal{D}) > \epsilon$, with probability at least $2/3$, using an expected number of $O\Big((\frac{\log(d)}{\alpha_1} + \frac{\log(d)}{\alpha_2})\max\{\sqrt{d}/\epsilon^2, d^{3/4}/\epsilon\}\Big)$ samples from $\mathcal{Q}_{S_1}$ and $\mathcal{D}_{S_2}$.*

3.2.4 Learning in Total Variation Distance

Using Algorithm 1, we can learn a Boolean product distribution $\mathcal{D}(\boldsymbol{p})$, within ϵ in total variation distance, using samples from its fat truncation. The following uses a standard analysis of the sample complexity of learning a Boolean product distribution (see e.g., (KLSU18)).

Corollary 3.2.8. *Let $\mathcal{D}(\boldsymbol{p})$ be a Boolean product distribution and let $\mathcal{D}_S$ be any α-fat truncation of $\mathcal{D}$. Then, for any $\epsilon, \delta > 0$, we can compute a Boolean product distribution $\widehat{\mathcal{D}}(\widehat{\boldsymbol{p}})$ such that $d_{\mathrm{TV}}(\mathcal{D},\widehat{\mathcal{D}}) \leq \epsilon$, with probability at least $1-\delta$, using $O(d\log(d)\log(d/\delta)/(\epsilon^2\alpha))$ samples from $\mathcal{D}_S$.*

Proof. We assume that $p_i \leq 1/2$ and that for all $i \in [d]$, $p_i \geq \epsilon/(8d)$. Both are without loss of generality. The former can be enforced by flipping 0 and 1. For the latter, we observe that there exists a distribution $\mathcal{D}'$ with $d_{\mathrm{TV}}(\mathcal{D},\mathcal{D}') \leq \epsilon/2$ that satisfies the assumption ($\mathcal{D}'$ can be obtained from $\mathcal{D}$ by adding uniform noise in each coordinate with probability $1-\frac{\epsilon}{4d}$, see also (CDKS17, Sec. 4.1)).

By Proposition 3.1.3, for any two Boolean product distributions $\mathcal{D}(\boldsymbol{p})$ and $\widehat{\mathcal{D}}(\widehat{\boldsymbol{p}})$ with parameter vectors $\boldsymbol{p}, \widehat{\boldsymbol{p}} \in (0,1)^d$, it holds that

$$d_{\mathrm{TV}}(\mathcal{D},\widehat{\mathcal{D}}) \leq \sqrt{2 \cdot \sum_{i=1}^{d} \frac{(p_i - \widehat{p}_i)^2}{(p_i + \widehat{p}_i)(2 - p_i - \widehat{p}_i)}}\,. \tag{3.2}$$

Similarly to the proof of Corollary 3.2.5, we take n samples $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n)}$ from Algorithm 1 and estimate each parameter p_i of $\mathcal{D}$ as $\widehat{p}_i = \sum_{\ell=1}^{n} y_i^{(\ell)}/n$. Using the Chernoff bound in (KLSU18, Claim 5.16), we show that for all directions $i \in [d]$, $\frac{(p_i-\widehat{p}_i)^2}{(p_i+\widehat{p}_i)(2-p_i-\widehat{p}_i)} \leq O(\log(d/\delta)/n)$. Drawing $n = O(d\log(d/\delta)/\epsilon^2)$ samples from Algorithm 1 and using Equation (3.2), we get that $d_{\mathrm{TV}}(\mathcal{D},\widehat{\mathcal{D}}) \leq O(\epsilon)$. The sample complexity follows from the fact that each sample of Algorithm 1 requires an expected number of $O(\log(d)/\alpha)$ samples from the α-fat truncation $\mathcal{D}_S$ of $\mathcal{D}$.

□

We can improve the sample complexity in Corollary 3.2.8, if the original distribution $\mathcal{D}$ is sparse. We say that a Boolean product distribution $\mathcal{D}(\boldsymbol{p})$ is (k, c)-*sparse*, for some $k \in [d]$ and $c \in [0, 1]$, if there is an index set $I \subseteq [d]$, with $|I| = d - k$, such that for all $i \in I$, $p_i = c$. Namely, we know that $d - k$ of $\mathcal{D}$'s parameters are equal to c (but we do not know which of them). Then, we first apply Corollary 3.2.5 and estimate all parameters of $\mathcal{D}$ within distance $\epsilon/\sqrt{k}$. We set each p_i with $|p_i - c| \leq \epsilon/\sqrt{k}$ to $p_i = c$. Thus, we recover the index set I. For the remaining k parameters, we apply Corollary 3.2.8. The result is summarized by the following:

Corollary 3.2.9. *Let $\mathcal{D}(\boldsymbol{p})$ be a (k, c)-sparse Boolean product distribution and let $\mathcal{D}_S$ be any α-fat truncation of $\mathcal{D}$. Then, for any $\epsilon, \delta > 0$, we can compute a Boolean product distribution $\widehat{\mathcal{D}}(\widehat{\boldsymbol{p}})$ such that $d_{\mathrm{TV}}(\mathcal{D}, \widehat{\mathcal{D}}) \leq \epsilon$, with probability at least $1 - \delta$, using $O\left(\frac{k \log(d) \log(d/\delta)}{\epsilon^2 \alpha}\right)$ samples from the truncated distribution $\mathcal{D}_S$.*

3.2.5 Learning Ranking Distributions from Truncated Samples

An interesting application of Theorem 3.2.3 is parameter estimation of ranking distributions from truncated samples. For clarity, we next focus on Mallows distributions. Our techniques imply similar results for other well known models of ranking distributions, such as Generalized Mallows distributions (FV86) and the models of (Pla75; Luc59), (BT52a) and (Bab50).

Definition and Notation. We start with some notation specific to this section. Let $\mathcal{S}_d$ be the symmetric group over the finite set of items $[d]$. Given a ranking $\pi \in \mathcal{S}_d$, we let $\pi(i)$ denote the position of item i in π. We say that i precedes j in π, denoted by $i \succ_\pi j$, if $\pi(i) < \pi(j)$. The Kendall tau distance of two rankings π and σ, denoted by $D_\tau(\pi, \sigma)$, is the number of discordant item pairs in π and σ. Formally,

$$D_\tau(\pi, \sigma) = \sum_{1 \leq i < j \leq d} \mathbf{1}\{(\pi(i) - \pi(j))(\sigma(i) - \sigma(j)) < 0\} \,. \tag{3.3}$$

The *Mallows model* (Mal57) is a family of ranking distributions parameterized by the *central ranking* $\pi_0 \in \mathcal{S}_d$ and the *spread parameter* $\phi \in [0, 1]$. Assuming the Kendall tau distance between rankings, the probability mass function is $\mathcal{M}(\pi_0, \phi; \pi) = \phi^{D_\tau(\pi_0, \pi)}/Z(\phi)$, where the normalization factor is $Z(\phi) = \prod_{i=1}^{d} \frac{1-\phi^i}{1-\phi}$. For a given Mallows distribution $\mathcal{M}(\pi_0, \phi)$, we denote $p_{ij} = \mathbf{Pr}_{\pi \sim \mathcal{M}}[i \succ_\pi j]$ the probability that item i precedes item j in a random sample from $\mathcal{M}$.

Truncated Mallows Distributions. We consider parameter estimation for a Mallows distribution $\mathcal{M}(\pi_0, \phi)$ with sample access to its truncation $\mathcal{M}_S$ by a subset $S \subseteq \mathcal{S}_d$. Then, $\mathcal{M}_S(\pi) = \mathcal{M}(\pi)/\mathcal{M}(S)$, for each $\pi \in S$, and $\mathcal{M}_S(\pi) = 0$, otherwise.

Next, we generalize the notion of fatness to truncated ranking distributions and prove the equivalent of Theorem 3.2.4 and Corollary 3.2.5.

For a ranking π, we let $\textsc{flip}(\pi, i, j)$ denote the ranking π' obtained from π with the items i and j swapped. Formally, $\pi'(\ell) = \pi(\ell)$, for all items $\ell \in [d] \setminus \{i, j\}$, $\pi'(j) = \pi(i)$ and $\pi'(i) = \pi(j)$. We say that a truncated Mallows distribution $\mathcal{M}_S$ is *α-fat for the pair* (i, j), if $\mathbf{Pr}_{\pi\sim\mathcal{M}_S}[\textsc{flip}(\pi, i, j) \in S] \geq \alpha$, for some $\alpha > 0$. A truncated Mallows distribution $\mathcal{M}_S(\pi_0, \phi)$ is α-fat, if $\mathcal{M}_S$ is α-fat for all pairs (i, j), and *neighboring α-fat*, if $\mathcal{M}_S$ is α-fat for all pairs (i, j) that occupy neighboring positions in the central ranking π_0, i.e., for all pairs (i, j) with $|\pi_0(i) - \pi_0(j)| = 1$.

Parameter Estimation and Learning of Mallows Distributions from Truncated Samples. We present Algorithm 3 that draws a sample from the truncated Mallows distribution $\mathcal{M}_S$ and updates a vector $\boldsymbol{q}$ with estimations $\widehat{p}_{ij} = q_{ij}/(q_{ij} + q_{ji})$ of the probability p_{ij} that item i precedes item j in a sample from the true Mallows distribution $\mathcal{M}$.

Algorithm 3 Update the estimate q_{ij} using one sample from $\mathcal{M}_S$

```
procedure SAMPLE($\mathcal{M}_S, \boldsymbol{q}$)                ▷ $\mathcal{M}_S$ is (neighboring) $\alpha$-fat.
    Draw sample $\pi \sim \mathcal{M}_S$
    for all $(i, j)$ such that FLIP$(\pi, i, j) \in S$ do      ▷ We assume oracle access to $\mathcal{M}_S$.
        if $i \succ_\pi j$ then
            $q_{ij} \leftarrow q_{ij} + 1$
        else
            $q_{ji} \leftarrow q_{ji} + 1$
    return $\boldsymbol{q}$
```

The vector $\boldsymbol{q}$ is initialized to 0 for all item pairs (i, j) and is updated through successive calls to Algorithm 3. For each sample $\pi \sim \mathcal{M}_S$, Algorithm 3 updates either q_{ij} or q_{ji} for all item pairs (i, j) such that $\textsc{flip}(\pi, i, j) \in S$. Thus, we can show the following:

Theorem 3.2.10. *Let $\mathcal{M}(\pi_0, \phi)$ be a Mallows distribution with $\pi_0 \in \mathcal{S}_d$ and $\phi \in [0, 1-\gamma]$, for some constant $\gamma > 0$, and let $\mathcal{M}_S$ be any neighboring α-fat truncation of $\mathcal{M}$. Then,*

(i) For any $\delta > 0$, we can learn the central ranking π_0, with probability at least $1 - \delta$, using an expected number of $O(\log(d)\log(d/\delta)/(\gamma^2\alpha))$ samples from $\mathcal{M}_S$.

(ii) Assuming that the central ranking π_0 is known, for any $\epsilon, \delta > 0$, we can compute an estimation $\widehat{\phi}$ of the spread parameter such that $|\phi - \widehat{\phi}| \leq O(\epsilon)$, with probability at least $1-\delta$, using an expected number of $O(\log(1/\delta)/(\epsilon^2\alpha))$ samples from $\mathcal{M}_S$.

(iii) For any $\epsilon, \delta > 0$, we can compute a Mallows distribution $\widehat{\mathcal{M}}(\pi_0, \widehat{\phi})$ so that

$$d_{\mathrm{TV}}(\mathcal{M}, \widehat{\mathcal{M}}) \leq O(\epsilon)\,,$$

with probability at least $1-\delta$, using an expected number of

$$O(\log(d)\log(d/\delta)/(\gamma^2\alpha) + d\log(1/\delta)/(\epsilon^2\alpha))$$

samples from $\mathcal{M}_S$.

The following is similar in spirit to Theorem 3.2.4. To estimate p_{ij}, we call Algorithm 3 as long as $q_{ij} + q_{ji} < \log(2/\delta)/\epsilon^2$. For the proof, we apply the argument used in the proof of Theorem 3.2.3.(i) and the Hoeffding bound used in the proof of Theorem 3.2.4.

Corollary 3.2.11. *Let $\mathcal{M}$ be a Mallows distribution and let $\mathcal{M}_S$ be any truncation of $\mathcal{M}$. If $\mathcal{M}_S$ is α-fat for pair (i,j), for any $\epsilon, \delta > 0$, we can compute an estimation $\widehat{p}_{ij}$ of the probability $p_{ij} = \mathbf{Pr}_{\pi\sim\mathcal{M}}[i \succ_\pi j]$ such that $|p_{ij} - \widehat{p}_{ij}| \leq \epsilon$, with probability at least $1-\delta$, using an expected number of $O(\log(1/\delta)/(\epsilon^2\alpha))$ samples from $\mathcal{M}_S$.*

We next give a detailed proof of Theorem 3.2.10, which shows how Algorithm 3 can efficiently estimate the parameters of (and learn in total variation distance) a Mallows distribution $\mathcal{M}$ using samples from any neighboring α-fat truncation $\mathcal{M}_S$ of $\mathcal{M}$.

Proof of Theorem 3.2.10. To prove (i), we use the fact that there is a bijective mapping from rankings in $\mathcal{S}_d$ to transitive tournaments on d nodes. So, we think of $\boldsymbol{q}$ as a directed graph G on d nodes, where there is an edge between i and j if $q_{ij} + q_{ji} \geq n$, for some n sufficiently large, which, for simplicity, will be determined at the end of the proof. The edge is from i to j, if $q_{ij} > q_{ji}$, and from j to i, otherwise. We keep calling Algorithm 3 until a directed path including all nodes (i.e., a total order) is formed in G. If a cycle is formed in G, before a total order appears, we discard $\boldsymbol{q}$ and start the algorithm from scratch.

Since $\mathcal{M}_S$ is neighboring α-fat, for any such pair (i,j) of neighboring items in π_0, the probability that a fresh sample $\pi \sim \mathcal{M}_S$ in Algorithm 3 increases $q_{ij} + q_{ji}$ is at least α (by the definition of neighboring α-fatness). Using exactly the same reasoning as in the proof of Theorem 3.2.3.(ii), we show that the expected number of samples before d edges appear in G is $O(n\log(d)/\alpha)$.

Let us fix any pair of items i and j such that $i \succ_{\pi_0} j$ and there is an edge between i and j in G. For simplicity, we assume that $q_{ij} + q_{ji} = n$. For sake of intuition, one may think of i and j as neighboring in π_0, but our analysis does not require so. We note that $\mathbb{E}[q_{ij}] = np_{ij}$ and $\mathbb{E}[q_{ji}] = np_{ji}$, and let $m_{ij} = p_{ij} - p_{ji}$. Working as in (CPS13, (1)), we can show that $m_{ij} \geq \frac{1+\phi}{1-\phi} = \Omega(\gamma)$ (see also (BFFSZ19, Theorem 12)). Therefore, $\mathbb{E}[q_{ij}] = n \cdot \frac{1+m_{ij}}{2}$ and $\mathbb{E}[q_{ji}] = n \cdot \frac{1-m_{ij}}{2}$.

A standard application of the Hoeffding bound shows that if $n = O(\log(d/\delta)/m_{ij}^2)$, $\mathbf{Pr}[q_{ij} \le n/2] \le \delta/d^2$. Therefore, assuming that an edge between i and j is present in G, the edge is directed from i to j (i.e., as in π_0) with probability at least $1-\delta/d^2$. Applying the union bound, we get that when we stop calling Algorithm 3, all edges present in G are as in π_0 with probability at least $1-\delta$.

We are ready to finish the proof of Item (i). Putting everything together, we get that after an expected number of $O(\log(d)\log(d/\delta)/(\alpha\gamma^2))$ samples from the truncated Mallows distribution $\mathcal{M}_S$, a total order consistent with π_0 is formed in G, with probability at least $1-\delta$. Increasing n by a constant factor makes the probability that a cycle appears in G polynomially small in d, which allows us to bound the expected number of samples from $\mathcal{M}_S$ before we find a total order in G by $O(\log(d)\log(d/\delta)/(\alpha\gamma^2))$.

For (ii), we assume that we know the central ranking π_0. For simplicity, we assume that $\pi_0 = (1, \dots, d)$. Then, as in Corollary 3.2.11, we can estimate the probability $p_{12} = \mathbf{Pr}_{\pi\sim\mathcal{M}}[1 \succ_\pi 2]$ such that $|p_{12} - \widehat{p}_{12}| \le \epsilon$, with probability at least $1-\delta$, using an expected number of $O(\log(1/\delta)/(\epsilon^2\alpha))$ samples from $\mathcal{M}_S$. Using $\widehat{p}_{12}$, we compute an estimation $\widehat{m}_{12} = 2\widehat{p_{12}} - 1$ of $m_{12} = 2p_{12} - 1$. It is straightforward to verify that $|p_{12} - \widehat{p}_{12}| \le \epsilon$ implies that $|m_{12} - \widehat{m}_{12}| \le \epsilon$. Working as in (CPS13, (1)), we show that for each pair of neighboring items i and $i+1$ in the central ranking π_0, $m_{i(i+1)} = \frac{1-\phi}{1+\phi}$. The reason is that for any ranking π and any pair of items i and $i+1$, with $i \succ_\pi i+1$, that are neighboring in π_0, swapping i and $i+1$ results in a ranking π' with $D_\tau(\pi', \pi_0) = D_\tau(\pi, \pi_0) + 1$. Our estimation of ϕ is $\widehat{\phi} = \frac{1-\widehat{m}_{12}}{1+\widehat{m}_{12}}$, where $|m_{12} - \widehat{m}_{12}| \le \epsilon$ implies that $|\phi - \widehat{\phi}| \le O(\epsilon)$.

Part (iii) follows from (i), (ii) and (BFFSZ19, Theorem 15). We can learn π_0 using the algorithm of (i) and an estimation $\widehat{\phi}$ of ϕ such that $|\widehat{\phi} - \phi| \le \epsilon/\sqrt{d}$ using the estimator of (ii), with an expected number of $O(d\log(1/\delta)/(\epsilon^2\alpha))$ samples from $\mathcal{M}_S$. (BFFSZ19, Theorem 15) shows that if $|\widehat{\phi} - \phi| \le \epsilon/\sqrt{d}$, then $d_{\mathrm{TV}}(\mathcal{M}(\pi_0, \phi), \widehat{\mathcal{M}}(\pi_0, \widehat{\phi}) \le O(\epsilon)$.

□

In this section, we focused on various implications of a truncation set being fat. We close this section with some comments about efficiently learning truncated Mallows models and performing e.g., identity testing when the α-fatness property does not hold true.

Let us recall the problem of learning truncated Mallows models. We will focus on estimating the central ranking assuming that the dispersion parameter is known. In this setting, there exists a central ranking π_0 and the learner observes i.i.d. samples from $\mathcal{M}_S(\pi_0, \phi)$. The goal is to efficiently estimate π_0. In the non-truncated setting, $\Theta(\log(d))$ samples are required. Under the fatness condition, we provided an $O(\log^2(d))$ sample algorithm. However, the fatness condition can be dropped but it may be still possible to retrieve the central ranking. Using the techniques of the upcoming sections, one could execute the Projected SGD approach (Section 3.4) and, under some structural conditions on the Boolean product distribution of dimension $O(d^2)$ and the truncation set (e.g., anti-concentration), recover the central ranking using $\mathrm{poly}(d)$ samples. However, it is not clear whether this reduction is optimal. It is an interesting question for future work to give the right characterization of learnability for truncated Mallows distributions.

In the task of identity testing of truncated Boolean product distributions, there exists a target distribution $\mathcal{D}^\star$ specified to the tester via its d success probabilities and the algorithm observes i.i.d. samples from the unknown truncated Boolean product distribution $\mathcal{D}_S$. The goal is to accept if $\mathcal{D} = \mathcal{D}^\star$ with probability $2/3$ and to reject if $d_{\mathrm{TV}}(\mathcal{D}, \mathcal{D}^\star) > \epsilon$ with probability $2/3$. We assume that the tester has membership oracle access to the set S (note that the truncated target $\mathcal{D}_S^\star$ cannot even be parsed efficiently by the tester since its size may be exponential in d). If the fatness condition fails but the conditions of Section 3.3 hold true, then one could still perform the SGD approach (Section 3.4), learn the distribution and hence perform identity testing using a polynomial number of samples. It is an interesting question whether one could efficiently perform identity testing from truncated samples without learning the distribution.

3.3 Efficient Learnability from Truncated Samples: Necessary Conditions

We next discuss necessary conditions for identifiability and efficient learnability of a Boolean product distribution from truncated samples. For Assumption 1 and Lemma 3.3.1, we recall that we can assume without loss of generality that S is normalized so that $\mathcal{D}_S(\mathbf{0}) > 0$.

Assumption 1. *For the truncated Boolean product distribution $\mathcal{D}_S$, $\mathcal{D}_S(\mathbf{0}) > 0$ (after possible normalization) and there are d linearly independent $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(d)} \in S$ with $\mathcal{D}_S(\boldsymbol{x}^{(j)}) > 0$, $j \in [d]$.*

The proof of Lemma 3.3.1 demonstrates that recovering $\boldsymbol{p}$ requires the solution to a linear system, similar to that in Footnote 4, which is solvable if and only if Assumption 1 holds.

Lemma 3.3.1. *A Boolean product distribution $\mathcal{D}(\boldsymbol{p})$ on Π_d is identifiable from its truncation $\mathcal{D}_S$ if and only if Assumption 1 holds.*

Proof. Let us assume that $\mathbf{0} \in S$ and there are d linearly independent vectors $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(d)} \in S$. We have that $\mathcal{D}(\mathbf{0}) = \prod_{i=1}^d (1 - p_i)$, and for each $j \in [d]$,

$$\prod_{i : x_i^{(j)} = 1} p_i \prod_{i : x_i^{(j)} = 0} (1 - p_i) = \mathcal{D}(\boldsymbol{x}^{(j)})\,. \tag{3.1}$$

However, the right-hand side of Equation (3.1) cannot be directly obtained from the truncated distribution $\mathcal{D}_S$. Hence, we normalize Equation (3.1), by dividing both sides by $\mathcal{D}_S(\mathbf{0})$, and get that

$$\prod_{i:x_i^{(j)}=1} \frac{p_i}{1-p_i} = \frac{\mathcal{D}(\boldsymbol{x}^{(j)})}{\mathcal{D}(\mathbf{0})} . \tag{3.2}$$

We observe that $\frac{\mathcal{D}(\boldsymbol{x}^{(j)})}{\mathcal{D}(\mathbf{0})} = \frac{\mathcal{D}_S(\boldsymbol{x}^{(j)})}{\mathcal{D}_S(\mathbf{0})}$, because for all $\boldsymbol{x} \in S$, $\mathcal{D}_S(\boldsymbol{x}) = \mathcal{D}(\boldsymbol{x})/\mathcal{D}(S)$. So, after normalization, the right-hand side of Equation (3.2) becomes a constant $q_j \overset{\text{def}}{=} \frac{\mathcal{D}_S(\boldsymbol{x}^{(j)})}{\mathcal{D}_S(\mathbf{0})} > 0$, for all $j \in [d]$.

Taking logarithms in Equation (3.2), we obtain that $\sum_{i:x_i^{(j)}=1} z_i = \log q_j$, where $z_i = \log \frac{p_i}{1-p_i}$, or equivalently $\boldsymbol{z}^T \boldsymbol{x}^{(j)} = \log q_j$. Since $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(d)}$ are linearly independent, the corresponding linear system with d equations and d unknowns has a unique solution. Solving the linear system $\left\{\boldsymbol{z}^T \boldsymbol{x}^{(j)} = \log q_j\right\}_{j\in[d]}$, we recover $\boldsymbol{z}$ and eventually $\boldsymbol{p}$.

The converse follows from the observation that solving a linear system as the one above is the only way to recover $\boldsymbol{p}$ from $\mathcal{D}_S$ (a linear system is the input to any potential solver from an information-theoretic viewpoint). Specifically, the only way to recover $\boldsymbol{p}$ from $\mathcal{D}_S$ is to solve the system consisting of Equation (3.1), for $j = 1, \ldots, d$, or some other equivalent system with d equations and $p_1, \ldots, p_d$ as unknowns. The only way to recover $\mathcal{D}(\boldsymbol{x})$ is to normalize Equation (3.1) by dividing by $\mathcal{D}(\boldsymbol{x}')$, for some $\boldsymbol{x}' \in S$ with $\mathcal{D}_S(\boldsymbol{x}') > 0$. We can assume without loss of generality that $\boldsymbol{x}' = \mathbf{0}$, since we can normalize S so that $\boldsymbol{x}'$ becomes $\mathbf{0}$. After normalizing by $\mathcal{D}_S(\mathbf{0})$ and taking logarithms in Equation (3.2), recovering $\boldsymbol{z}$ and $\boldsymbol{p}$ requires a collection of d linearly independent equations, which correspond to d linearly independent $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(d)} \in S$ with $\mathcal{D}_S(\boldsymbol{x}^{(j)}) > 0$, for each $j \in [d]$. Technically, if Assumption 1 does not hold, the input contains a matrix with rank $< d$ and hence the true $\boldsymbol{p}$ is not uniquely identifiable.

□

We proceed to show two necessary conditions for *efficient learnability*. Our first condition is that we have oracle access to the truncation set S. More formally, we assume that:

Assumption 2. *S is accessible through a membership oracle, which reveals whether $\boldsymbol{x} \in S$, for any $\boldsymbol{x} \in \Pi_d$.*

Based on the proof of (DGTZ18, Lemma 12), we show that if Assumption 2 does not hold, we can construct a (possibly exponentially large) truncation set S so that $\mathcal{D}_S$ appears identical to the uniform distribution $\mathcal{U}$ on Π_d as long as all the samples are distinct.

Lemma 3.3.2. *For any Boolean product distribution $\mathcal{D}(\boldsymbol{p})$, there is a truncation set S so that without additional information about S, we cannot distinguish between sampling from $\mathcal{D}_S$ and sampling from the uniform distribution $\mathcal{U}$ on Π_d, before an expected number of $\Omega(\sqrt{|S|})$ samples are drawn.*

Proof. The truncation set $S = S_1 \times \cdots \times S_d$ is the product of d truncation sets S_i, one in each direction $i \in [d]$. If $p_i \geq 1/2$, $S_i = \{0,1\}$ with probability $\frac{1-p_i}{p_i}$, and $S_i = \{0\}$, otherwise. If $p_i < 1/2$, $S_i = \{0,1\}$ with probability $\frac{p_i}{1-p_i}$, and $S_i = \{1\}$, otherwise. There is a constant $c > 0$ such that if $|p_i - 1/2| \leq c$, for all $i \in [d]$, $|S|$ is exponential in d with constant probability.

By the principle of deferred decisions, we can think of the sampling process from $\mathcal{D}_S$ as follows: we draw a sample $\boldsymbol{x} \sim \mathcal{D}$. If this is the first time that $\boldsymbol{x}$ is drawn from $\mathcal{D}$, for each $i \in [d]$, independently, x_i survives with probability $\min\{\mathcal{B}e(p_i; 1-x_i)/\mathcal{B}e(p_i; x_i), 1\}$. If every x_i survives, $\boldsymbol{x}$ is added to S and becomes a sample from $\mathcal{D}_S$. If $\boldsymbol{x}$ has been drawn before, $\boldsymbol{x}$ becomes a sample from $\mathcal{D}_S$ if and only if $\boldsymbol{x} \in S$, so that new samples are treated consistently with past ones.

We note that as long as a duplicate sample does not appear, the probability that $x_i = 0$ and x_i survives is equal to the probability that $x_i = 1$ and x_i survives, for all $i \in [d]$. In fact, the following process samples from the uniform distribution $\mathcal{U}_d$ on Π_d: we draw a sample $\boldsymbol{x} \sim \mathcal{D}$. Then, for each $i \in [d]$, independently, x_i survives with probability $\min\{\mathcal{B}e(p_i; 1-x_i)/\mathcal{B}e(p_i; x_i), 1\}$. If every x_i survives, $\boldsymbol{x}$ is returned as a sample from $\mathcal{U}_d$. The difference is that there is no truncation set. So, we do not need to treat new samples consistently with past ones.

Before the first duplicate sample is drawn from $\mathcal{D}_S$, there is no way to distinguish between sampling from $\mathcal{D}_S$ and sampling from $\mathcal{U}_d$. By the birthday problem, the appearance of the first duplicate sample from $\mathcal{D}_S$ requires an expected number of $\Omega(\sqrt{|S|})$ samples from $\mathcal{D}_S$.

We highlight that we can easily distinguish between sampling from $\mathcal{D}_S$ and sampling from $\mathcal{U}$, if we have oracle access to the truncation set S.

□

Our second necessary condition for efficient learnability is that the truncated distribution is not extremely well concentrated in any direction. Intuitively, we need the Boolean product distribution $\mathcal{D}$, and its truncation $\mathcal{D}_S$, to behave well, so that we can get enough information about $\mathcal{D}$ based on few samples from $\mathcal{D}_S$. More formally, we quantify $\mathcal{D}_S$'s anti-concentration using $\lambda^\star$, which is the maximum positive number so that for all unit vectors $\boldsymbol{w} \in \mathbb{R}^d$, $\|\boldsymbol{w}\|_2 = 1$, and all $c \in \mathbb{R}$, $\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{D}_S}[\boldsymbol{w}^T\boldsymbol{x} \notin (c - \lambda^\star, c + \lambda^\star)] \geq \lambda^\star$. Assumption 3 requires that $\lambda^\star$ is polynomially large in $1/d$.

Assumption 3. *There exists a $\lambda \geq 1/\mathrm{poly}(d)$ such that for all unit vectors $\boldsymbol{w} \in \mathbb{R}^d$, $\|\boldsymbol{w}\|_2 = 1$, and all $c \in \mathbb{R}$, $\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{D}_S}[\boldsymbol{w}^T\boldsymbol{x} \notin (c - \lambda, c + \lambda)] \geq \lambda$.*

We note that Assumption 3 is a stronger version of Assumption 1. It also implies that all parameters $p_i \in (0,1)$ are bounded away from 0 and 1 by a safe margin (we will focus on parameters whose margin from 0 and 1 is dimension-independent). We next show that if $\mathcal{D}_S$ is well concentrated in some direction, estimating the parameter vector $\boldsymbol{p}$ requires a large number of samples from $\mathcal{D}_S$. More specifically, we show that either estimating $\mathcal{D}_S(\mathbf{0})$, which is needed for normalizing the linear system in Lemma 3.3.1, or sampling d vectors that result in a well-conditioned linear system, require $\Omega(1/\lambda^\star)$ samples from $\mathcal{D}_S$. Therefore, if Assumption 3 does not hold, estimating $\boldsymbol{p}$ with truncated samples from $\mathcal{D}_S$ has superpolynomial sample complexity.

Lemma 3.3.3. *Assume that Assumption 3 does not hold true, i.e., the optimal anti-concentration parameter $\lambda^\star$ satisfies $1/\lambda^\star = \omega(\mathrm{poly}(d))$. Let $\mathcal{D}(\boldsymbol{p})$ be a Boolean product distribution and let $\mathcal{D}_S$ be a truncation of $\mathcal{D}$. Then, computing an estimation $\widehat{\boldsymbol{p}}$ of the parameter vector $\boldsymbol{p}$ of $\mathcal{D}$ such that $\|\boldsymbol{p}-\widehat{\boldsymbol{p}}\|_2 \leq o(1)$ requires an expected number of $\Omega(1/\lambda^\star)$ samples from $\mathcal{D}_S$.*

Let us first provide some intuition. For a unit vector $\boldsymbol{w} \in \mathbb{R}^d$, we think of the space $H_{\boldsymbol{w}} = \{\boldsymbol{x} \in S : \boldsymbol{w}^T\boldsymbol{x} \in (c-\lambda, c+\lambda)\}$. If $\lambda^\star$ is very small, there is a direction $\boldsymbol{w}$ such that virtually all samples $\boldsymbol{x} \sim \mathcal{D}_S$ lie in $H_{\boldsymbol{w}}$. Intuitively, recovering ($\boldsymbol{z}$ and) $\boldsymbol{p}$ boils down to the solution of a linear system as that in Footnote 4 and in Lemma 3.3.1. For that, we need d linearly independent vectors $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(d)} \in S$ and an additional fixed element $\boldsymbol{x}^\star \in S$ for the normalization of the probabilities in the right-hand side. With high probability, all $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(d)} \in H_{\boldsymbol{w}}$. If $\boldsymbol{x}^\star$ is also in $H_{\boldsymbol{w}}$, normalizing the system by $\boldsymbol{x}^\star$ results in an ill-conditioned system. In fact, we can show that the condition number of the system is $\Omega(1/\lambda^\star)$. Therefore, solving the linear system efficiently requires sampling a vector $\boldsymbol{x}^\star \notin H_{\boldsymbol{w}}$ for normalization. However, the probability that we sample (and thus, can use for normalization) a vector $\boldsymbol{x}^\star \notin H_{\boldsymbol{w}}$ is at most $\lambda^\star$.

We now proceed with the proof of Lemma 3.3.3.

Proof. Next, we formalize the intuition behind the sketch of the proof. We recall that for a fixed unit vector $\boldsymbol{w} \in \mathbb{R}^d$, we let $H_{\boldsymbol{w}} = \{\boldsymbol{x} \in S : \boldsymbol{w}^T\boldsymbol{x} \in (c-\lambda, c+\lambda)\}$. By the definition of λ^*, for any $\lambda > \lambda^*$, there is a unit vector $\boldsymbol{w} \in \mathbb{R}^d$ and a $c \in \mathbb{R}$ such that $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}_S}[\boldsymbol{x} \notin H_{\boldsymbol{w}}] < \lambda$, or equivalently, $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}_S}[\boldsymbol{x} \in H_{\boldsymbol{w}}] \geq 1-\lambda$.

We recall that we assume without loss of generality that S is normalized so that $\mathbf{0} \in S$ and $\mathcal{D}_S(\mathbf{0}) > 0$. In fact, $\mathbf{0}$ plays the role of the fixed element $\boldsymbol{x}^\star$, discussed in the sketch, which we use for normalization. Next, we distinguish between two cases based on whether $\mathbf{0} \in H_{\boldsymbol{w}}$ or not.

Let us first fix $\lambda > \lambda^\star$ that lies in a small neighborhood of $\lambda^\star$ of radius ϵ, where ϵ is sufficiently small. We will show that for any such λ (that satisfies that $1/\lambda$ is (almost) super-polynomial in d), we get a sample complexity of order $1/\lambda$. Since this property will hold arbitrarily close to $\lambda^\star$, the sample complexity will be super-polynomial in the dimension d.

Having chosen λ as above, there is a direction $\boldsymbol{w}$ and a translation $c \in \mathbb{R}$, that define the space $H_{\boldsymbol{w}}$, such that $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{D}_S}[\boldsymbol{x} \notin H_{\boldsymbol{w}}] < \lambda$. There are two cases for the translation c.

Case A: We may first assume that c is small enough, that is $|c| < \lambda$ and, hence, $0 \in (c-\lambda, c+\lambda)$. Let X be any set of $O(1/\lambda)$ samples from $\mathcal{D}_S$. Then, with constant probability, all $X \subseteq H_{\boldsymbol{w}}$. Let $\boldsymbol{X}_d = [\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(d)}]^T$ be the matrix obtained by any

d elements $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(d)} \in X$ different from $\mathbf{0}$. By Lemma 3.3.1, recovering $\boldsymbol{p}$ requires the solution of the linear system $\boldsymbol{X}_d \boldsymbol{z} = \log(\boldsymbol{q})$, where $\log(\boldsymbol{q}) = (\log(q_j))_{j\in[d]}$ and $q_j = \frac{\mathcal{D}_S(\boldsymbol{x}^{(j)})}{\mathcal{D}_S(\mathbf{0})}$, for each $j \in [d]$.

We next show that since $c \in (-\lambda, \lambda)$, with constant probability, the matrix $\boldsymbol{X}_d$ is ill-conditioned and has condition number [6] $\kappa(\boldsymbol{X}_d) = \Omega(1/\lambda)$.

Specifically, since all $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(d)}$ are different from $\mathbf{0}$, there is a unit vector $\boldsymbol{w'} \in \mathbb{R}^d$ so that $\|\boldsymbol{X}_d \boldsymbol{w'}\|_2 \geq 1$. On the other hand, by the hypothesis that with constant probability, $X \subseteq H_{\boldsymbol{w}}$, $\|\boldsymbol{X}_d \boldsymbol{w}\|_2 \leq (|c| + \lambda) \cdot \sqrt{d} \leq 2\lambda \cdot \sqrt{d}$. Therefore, the condition number of the matrix $\boldsymbol{X}_d$ is $\kappa(\boldsymbol{X}_d) = \Omega(1/(\lambda \cdot \sqrt{d}))$ for the fixed $\lambda > \lambda^*$ in the neighborhood of $\lambda^\star$. This implies that the condition matrix is of order $\Omega(1/\lambda)$. Hence, with constant probability, we cannot recover ($\boldsymbol{z}$ and) $\boldsymbol{p}$ within accuracy $o(1)$, unless we estimate the right-hand side $\boldsymbol{q}$ of the linear system $\boldsymbol{X}_d \boldsymbol{z} = \log(\boldsymbol{q})$ with accuracy $o(\lambda)$, which requires $\omega(1/\lambda)$ samples.

Case B: Otherwise, if $|c| > \lambda$, then $0 \notin (c - \lambda, c + \lambda)$. Since $\boldsymbol{w}^T \mathbf{0} = 0$, the probability that $\mathbf{0}$ is sampled from $\mathcal{D}_S$ is at most λ. Hence, unless we take $\omega(1/\lambda)$ samples, we cannot find a good estimation of $\mathcal{D}_S(\mathbf{0})$, which is required for the linear system $\boldsymbol{X}_d \boldsymbol{z} = \log(\boldsymbol{q})$, whose solution recovers ($\boldsymbol{z}$ and) $\boldsymbol{p}$.

Finally, since either Case A or B will hold for any $\lambda > \lambda^*$ in the ϵ-neighborhood of $\lambda^\star$, we let $\lambda \downarrow \lambda^*$ and hence we get that an expected number of $\Omega(1/\lambda^*)$ samples is required, which is super-polynomial in d.

□

The above condition highlights a gap between the continuous problem of learning truncated Gaussian distributions (DGTZ18) and the discrete case, where truncation can be quite restrictive.

For the efficient estimation of $\boldsymbol{z}$, we also need to assume that the truncation set S is large enough. Namely, we assume that:

Assumption 4. *For the truncation set S, there is a constant $\alpha > 0$ so that the Boolean product distribution $\mathcal{D}$ has $\mathcal{D}(S) \geq \alpha$.*

Assumption 4 is not necessary for efficient learning, in the sense that e.g., there may be α-fat product distributions which do not satisfy this condition, but are still efficiently learnable using Corollary 3.2.8.

We conclude this section with a remark. Note that complex models, such as Bayes networks and Ising models, can be cast as truncated product distributions in a Boolean hypercube of appropriately high dimension (the translation is conceptually similar to that for Mallows models in Section 3.2.5). For instance, the Ising model over $\{-1, +1\}^d$ with interaction matrix J (with $J_{ii} = 0$) and external field h is defined by normalizing the function $\pi(x) = \langle x, Jx \rangle + h^T x$. We have a dimension

[6] Let $\boldsymbol{A}$ be a $d \times d$ square matrix with singular values $s_1 \geq \cdots \geq s_d \geq 0$. We will denote with $s_{\max}(\boldsymbol{A}) = s_1$ and with $s_{\min}(\boldsymbol{A}) = s_d$. The condition number of the $\boldsymbol{A}$ is $\kappa(\boldsymbol{A}) = s_{\max}(\boldsymbol{A})/s_{\min}(\boldsymbol{A})$. The condition number $\kappa(\boldsymbol{A}) \in [1, \infty]$ quantifies the sensitivity of the solution to a linear system $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}$ to the small perturbations of $\boldsymbol{b}$.

for each edge and a dimension for each spin (so the Boolean Product distribution is a measure over $\{-1,+1\}^{\binom{d}{2}+d}$) and the truncation set S_{Ising} consists of all 2^d binary vectors with valid edge labels (i.e., vectors with edge labels consistent with some allocation of $\{+,-\}$ to the vertices). Hence, we can consider the product probability measure over the points $x \in \{-1,+1\}^{\binom{d}{2}+d}$ with density

$$\mathcal{D}(x) = \mathcal{D}((x_{uv})_{u,v\in E}, (x_u)_{u\in V}) = \prod_{(u,v)\in E} \frac{\exp(J_{uv}x_{uv})}{2\cosh(J_{uv})} \prod_{u\in V} \frac{\exp(h_u x_u)}{2\cosh(h_u)}\,.$$

Casting an Ising model μ to our setting results in a truncated Boolean product distribution $\mu(x) = \mathcal{D}(x)\mathbf{1}\{x \in S\}/\mathcal{D}(S_{\text{Ising}})$ that satisfies Assumption 1, Assumption 2 and Assumption 3 (assuming that the parameters of the Ising model are "sufficiently nice" so that the $\binom{d}{2}+d$ parameters of $\mathcal{D}$ are bounded away from 0 and 1), but it is not guaranteed to satisfy Assumption 4 (which is in accordance with the fact that sampling from an Ising model is computationally hard in general (see e.g., (Hub99))).

In the following section, we present the Projected Stochastic Gradient Descent algorithm and show that assumptions 2, 3 and 4 are sufficient for the efficient estimation of the natural parameter vector $\boldsymbol{z}$ of the Boolean product distribution $\mathcal{D}$ by sampling from its truncation $\mathcal{D}_S$.

3.4 PSGD for Learning Truncated Boolean Product Distributions

We next show how to estimate efficiently the natural parameter vector $\boldsymbol{z}^\star$ of a Boolean product distribution $\mathcal{D}(\boldsymbol{z}^\star)$ using samples from its truncation $\mathcal{D}_S(\boldsymbol{z}^\star)$, assuming that the true distribution satisfies the conditions 2, 3 and 4.

Similarly to (DGTZ18), we use Projected Stochastic Gradient Descent (SGD) on the negative log-likelihood of the truncated samples. Our SGD algorithm is described in Algorithm 4. We should highlight that Algorithm 4 runs in the space of the natural parameters $\boldsymbol{z}$ of the Boolean product distribution. Changing the parameters from $\boldsymbol{p}$ to $\boldsymbol{z}$ results in a linear system, similar to that in Footnote 4 and in the proof of Lemma 3.3.1 and simplifies the analysis of the log-likelihood function. Furthermore, by Proposition 3.1.3, estimating $\boldsymbol{z}^\star$ within error at most ϵ in L_2 norm results in a distribution within total variation distance at most ϵ to $\mathcal{D}(\boldsymbol{z}^\star)$.

Throughout the analysis of Algorithm 4, we make use of Assumptions 2 - 4. The technical details of the analysis are deferred to Section 3.4.1. The analysis goes as follows: we first derive the negative log-likelihood function that Algorithm 4 optimizes. Since the truncation set S is only accessed through membership queries, we do not have a closed form of the log-likelihood.

Algorithm 4 Projected Stochastic Gradient Descent with Samples from $\mathcal{D}_S(\boldsymbol{p}^\star)$.

```
procedure SGD($M, \eta$)                         ▷ $M$ : number of steps, $\eta$ : parameter
    $\boldsymbol{z}^{(0)} \leftarrow \widehat{\boldsymbol{z}}$                  ▷ $\widehat{\boldsymbol{z}}$ is the empirical estimate of Lemma 3.5.3.
    for $t = 1..M$ do
        Sample $\boldsymbol{x}^{(t)}$ from $\mathcal{D}_S$
        repeat
            Sample $\boldsymbol{y}$ from $\mathcal{D}(\boldsymbol{z}^{(t-1)})$
        until $\boldsymbol{y} \in S$                      ▷ We assume oracle access to $S$.
        $\boldsymbol{v}^{(t)} \leftarrow -\boldsymbol{x}^{(t)} + \boldsymbol{y}$
        $\boldsymbol{z}^{(t)} \leftarrow \Pi_{\mathcal{B}}(\boldsymbol{z}^{(t-1)} - \frac{1}{t \cdot \eta}\boldsymbol{v}^{(t)})$          ▷ $\eta_t = 1/(t \cdot \eta)$: step size
    return $\overline{\boldsymbol{z}} \leftarrow \frac{1}{M}\sum_{t=1}^{M} \boldsymbol{z}^{(t)}$
```

However, we can show that it is convex for any truncation set S. We prove that the natural parameter vector $\widehat{\boldsymbol{z}}$ corresponding to the empirical estimate $\widehat{\boldsymbol{p}}_S$ is a good initialization for Algorithm 4. Specifically, we show that $\widehat{\boldsymbol{p}}_S$ is close to the true parameter vector $\boldsymbol{p}^\star$ in L_2 distance, and that this proximity holds for the corresponding natural parameter vectors as well.

For the correctness of Algorithm 4, it is essential that it runs in a convex region. We can show that there exists a ball $\mathcal{B}$, centered at the initialization point $\widehat{\boldsymbol{z}}$, which contains $\boldsymbol{z}^\star$. The radius of the ball depends only on the lower bound α of $\mathcal{D}(S)$ (Assumption 4). We can prove that Assumptions 3 and 4 always hold inside $\mathcal{B}$. That is, for any vector $\boldsymbol{z} \in \mathcal{B}$ (and the corresponding parameter vector $\boldsymbol{p}$), the anti-concentration assumption holds for $\mathcal{D}_S(\boldsymbol{p})$ and the mass assigned to the truncation set S by $\mathcal{D}_S(\boldsymbol{p})$ can be lower bounded by a polynomial function of α.

Under these two assumptions, we can prove that the negative log-likelihood is strongly-convex inside the ball $\mathcal{B}$. Hence, while Algorithm 4 iterates inside $\mathcal{B}$, the truncation set has always constant mass and the negative log-likelihood remains strongly-convex. Consequently, Algorithm 4 converges to the true vector of natural parameters $\boldsymbol{z}^\star$. The following theorem is the main result of the steps described above. For the next result, recall that we consider a target Boolean product distribution $\mathcal{D}(\boldsymbol{p}^\star)$ whose parameters' margin from 0 and 1 is dimension-independent.

Theorem 3.4.1. *Given oracle access to a measurable set $S \subseteq \Pi_d$ (Assumption 2), whose measure under some unknown Boolean product distribution $\mathcal{D}(\boldsymbol{z}^\star)$ is at least some constant $\alpha > 0$ (Assumption 4) and where the truncated distribution $\mathcal{D}_S(\boldsymbol{z}^\star)$ satisfies Assumption 3 with parameter λ, and given samples from the truncation $\mathcal{D}_S(\boldsymbol{z}^\star)$, there exists a polynomial-time algorithm that recovers an estimation $\overline{\boldsymbol{z}}$ of $\boldsymbol{z}^\star$. For any $\epsilon > 0$, the algorithm uses $\mathrm{poly}(1/\lambda)\widetilde{O}(d/\epsilon^2)$ truncated samples from $\mathcal{D}_S(\boldsymbol{z}^\star)$ and membership queries to S and guarantees that $\|\boldsymbol{z}^\star - \overline{\boldsymbol{z}}\|_2 \leq \epsilon$, with* *probability 99%. Under these conditions, it also holds that* $d_{\mathrm{TV}}(\mathcal{D}(\boldsymbol{z}^\star), \mathcal{D}(\overline{\boldsymbol{z}})) \leq O(\epsilon)$.

3.4.1 Projected SGD: Algorithm's Description

In this section, we present and explain the Projected SGD algorithm that learns the true natural parameter vector $\boldsymbol{z}^\star$ and, consequently, as we showed in Proposition 3.1.3, learns the true Boolean product distribution $\mathcal{D}(\boldsymbol{p}^\star)$ in total variation distance.

We are now ready to present the main steps of our SGD Algorithm 4. The input of the algorithm is the number of the steps M and a parameter η, that modifies the step size. The initialization point $\boldsymbol{z}^{(0)}$ of the algorithm will be the point $\widehat{\boldsymbol{z}}$, that equals to the natural parameter vector of the empirical estimate $\widehat{\boldsymbol{p}}_S$, defined by Equation (3.3). For $t \in [M]$, our guess for the true natural parameter vector $\boldsymbol{z}^\star$ will be denoted by $\boldsymbol{z}^{(t)}$. In each round t, we produce a guess $\boldsymbol{z}^{(t)}$ as follows: Firstly, we draw a sample $\boldsymbol{x}^{(t)}$ from the unknown truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$. Also, we draw a second sample $\boldsymbol{y}$ from the distribution induced by our previous guess $\boldsymbol{z}^{(t-1)}$. Note that it is possible that the generated sample $\boldsymbol{y}$ does not lie in the truncation set S. Hence, we have to iterate until we draw a sample that lies in S, that is $M_S(\boldsymbol{y}) = \mathbf{1}_{\boldsymbol{y}\in S}$ is equal to 1. As we have already mentioned, the function that we are minimizing is the negative log-likelihood for the population model. As we will see in Lemma 3.5.1 and Equation (3.2), the true gradient of this function is equal to

$$-\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_S(\boldsymbol{z}^\star)}[\boldsymbol{x}] + \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{D}_S(\boldsymbol{z})}[\boldsymbol{y}] \,.$$

In Algorithm 4, this quantity corresponds to a random direction denoted by $\boldsymbol{v}^{(t)}$ at step t and is equal to $-\boldsymbol{x}^{(t)} + \boldsymbol{y}$. Note that its expected value is equal to the true gradient. Hence, as in the classical gradient descent setting, we update our guess using the following update rule

$$\boldsymbol{z}^{(t)} \leftarrow \boldsymbol{z}^{(t-1)} - \eta_t \boldsymbol{v}^{(t)} \,.$$

As we have explained, we perform the SGD algorithm in a ball $\mathcal{B}$ of radius. Hence, it may be the case that our new guess $\boldsymbol{z}^{(t)}$ lies outside $\mathcal{B}$. Hence, we have to project that point back to the ball. For that reason, we use the projection function $\Pi_{\mathcal{B}}$, that equals to the mapping

$$\Pi_{\mathcal{B}}(\boldsymbol{x}) = \mathop{\mathrm{argmin}}_{\boldsymbol{z}\in\mathcal{B}} \|\boldsymbol{x} - \boldsymbol{z}\|_2 \text{ for } \boldsymbol{x} \in \mathbb{R}^d \,.$$

Finally, after M steps, the SGD algorithm returns an estimate $\overline{\boldsymbol{z}}$ that is close to the minimizer of the negative log-likelihood function. As we will show, this minimizer corresponds to the true natural parameters vector $\boldsymbol{z}^\star$. In the next section, we perform the theoretical analysis of the projected stochastic gradient descent algorithm for truncated Boolean product distributions.

3.5 Projected SGD: Theoretical Analysis

Our goal is to prove Theorem 3.4.1. The roadmap of the proof is presented as follows:

- **Convexity of the objective.** In Section 3.5.1, we show that the population version of the negative log-likelihood objective is convex with respect to the natural parameter vector (see Lemma 3.5.1 and Section 3.5.1).
- **Initial feasible point.** In Section 3.5.2, we efficiently compute a good initialization point for the SGD algorithm. The statement is presented in Lemma 3.5.3.
- **Feasible region.** In Section 3.5.3, we show that there exists a ball (and hence an easy-to-project set) that contains the true vector $\boldsymbol{z}^\star$ (see Lemma 3.5.6) and each point in the ball satisfies Assumptions 3 (see Lemma 3.5.9) and 4 (see Lemma 3.5.7).
- **Unbiased estimation of the gradient.** In Section 3.5.4, we show how to obtain an unbiased estimation of the gradient of the objective efficiently.
- **Strong convexity inside the feasible region.** In Section 3.5.5, we establish that the negative log-likelihood objective is strongly-convex inside the ball of Section 3.5.3.
- **Analysis of the SGD algorithm.** In Section 3.5.6, we show that the bounded variance step property holds (see Lemma 3.5.13). Hence, combining this result with the strong-convexity inside the ball, we can apply Fact 1 and get Theorem 3.4.1.

3.5.1 Convexity of the negative log-likelihood

Let S be a subset of the hypercube Π_d and $\mathcal{D}(\boldsymbol{p})$ be an arbitrary Boolean product distribution. We remind the reader that, for $\boldsymbol{x} \in \Pi_d$:

$$\mathcal{D}(\boldsymbol{p}; \boldsymbol{x}) = \mathcal{B}e(p_1; x_1) \otimes \cdots \otimes \mathcal{B}e(p_d; x_d) = \prod_{i \in [d]} \left(p_i^{x_i}(1-p_i)^{1-x_i}\right).$$

Let $\boldsymbol{z}$ be the natural parameters vector with $z_i = \log \frac{p_i}{1-p_i}$ for $i \in [d]$. Rewriting the distribution as an exponential family, we get that:

$$\mathcal{D}(\boldsymbol{p}; \boldsymbol{x}) = \prod_{i \in [d]} \exp\left(x_i \log \frac{p_i}{1-p_i} + \log(1-p_i)\right),$$

or equivalently:

$$\mathcal{D}(\boldsymbol{z}; \boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^T \boldsymbol{z})}{\prod_{i \in [d]}(1 + \exp(z_i))}.$$

The truncation set S induces a distribution $\mathcal{D}_S(\boldsymbol{z})$, that is equal to:

$$\mathcal{D}_S(\boldsymbol{z};\boldsymbol{x}) = \mathbf{1}_{\boldsymbol{x}\in S}\frac{\exp(\boldsymbol{x}^T\boldsymbol{z})}{\sum_{\boldsymbol{y}\in S}\exp(\boldsymbol{y}^T\boldsymbol{z})}\,.$$

Afterwards, we compute the negative log-likelihood $\ell(\boldsymbol{z})$ of the truncated samples drawn from the truncated distribution $\mathcal{D}_S(\boldsymbol{z})$ and study its behavior in terms of convexity.

Log-likelihood for a Single Sample

Notice that the structure of the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{z})$, expressed as an exponential family, is quite useful when computing the negative log-likelihood for a single sample $\boldsymbol{x}$ drawn from a distribution $\mathcal{D}_S(\boldsymbol{z})$, that is:

$$\ell(\boldsymbol{z};\boldsymbol{x}) = -\log\mathcal{D}_S(\boldsymbol{z};\boldsymbol{x}) = -\boldsymbol{x}^T\boldsymbol{z} + \log\left(\sum_{\boldsymbol{y}\in S} e^{\boldsymbol{y}^T\boldsymbol{z}}\right). \tag{3.1}$$

The convexity of the negative log-likelihood $\ell(\boldsymbol{z})$ of the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{z})$ follows immediately if one computes the gradient and the Hessian of $\ell(\boldsymbol{z})$ with respect to the natural parameter vector $\boldsymbol{z}$. This result is presented in the following Lemma.

Lemma 3.5.1. *The negative log-likelihood objective $\ell(\boldsymbol{z};\boldsymbol{x})$, as defined in Equation (3.1), is convex with respect to $\boldsymbol{z}$ for all $\boldsymbol{x}\in\Pi_d$.*

Proof. Observe that the negative log-likelihood of a single sample $x\sim\mathcal{D}_S(\boldsymbol{z})$ will be

$$\ell(\boldsymbol{z};\boldsymbol{x}) = -\boldsymbol{x}^T\boldsymbol{z} + \log\left(\sum_{\boldsymbol{y}\in S} e^{\boldsymbol{y}^T\boldsymbol{z}}\right).$$

We now compute the gradient of $\ell(\boldsymbol{z};\boldsymbol{x})$ with respect to the parameter $\boldsymbol{z}$.

$$\nabla_{\boldsymbol{z}}\ell(\boldsymbol{z};\boldsymbol{x}) = -\boldsymbol{x} + \frac{\sum_{\boldsymbol{y}\in S}\boldsymbol{y}e^{\boldsymbol{y}^T\boldsymbol{z}}}{\sum_{\boldsymbol{y}\in S}e^{\boldsymbol{y}^T\boldsymbol{z}}} = -\boldsymbol{x} + \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{D}_S(\boldsymbol{z})}[\boldsymbol{y}]\,.$$

Finally, we compute the Hessian of the negative log-likelihood:

$$\boldsymbol{H}_\ell(\boldsymbol{z}) = \frac{\sum_{\boldsymbol{y}\in S}\boldsymbol{y}\boldsymbol{y}^Te^{\boldsymbol{y}^T\boldsymbol{z}}}{\sum_{\boldsymbol{y}\in S}e^{\boldsymbol{y}^T\boldsymbol{z}}} - \frac{\sum_{\boldsymbol{y}\in S}\boldsymbol{y}e^{\boldsymbol{y}^T\boldsymbol{z}}}{\sum_{\boldsymbol{y}\in S}e^{\boldsymbol{y}^T\boldsymbol{z}}}\frac{\sum_{\boldsymbol{y}\in S}\boldsymbol{y}e^{\boldsymbol{y}^T\boldsymbol{z}}}{\sum_{\boldsymbol{y}\in S}e^{\boldsymbol{y}^T\boldsymbol{z}}} = \mathrm{Cov}_{\boldsymbol{y}\sim\mathcal{D}_S(\boldsymbol{z})}[\boldsymbol{y},\boldsymbol{y}]\,.$$

The Hessian of the negative log-likelihood $\boldsymbol{H}_\ell$ is semi-positive definite since it equals to a covariance matrix (in particular, it equals to the covariance matrix of the sufficient statistics of the exponential family). The result follows. □

Log-likelihood for the Population Model

Our Projected SGD algorithm will optimize the negative log-likelihood for the population model, that will be denoted with $\bar{\ell}$. This function is defined as the expected value of the negative log-likelihood function with respect to the true truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{z}^\star)$, that is

$$\bar{\ell}(\boldsymbol{z}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_S(\boldsymbol{z}^\star)} [\ell(\boldsymbol{z};\boldsymbol{x})] \,.$$

Using the formula of Equation (3.1), we get that

$$\bar{\ell}(\boldsymbol{z}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_S(\boldsymbol{z}^\star)} \left[-\boldsymbol{x}^T\boldsymbol{z} + \log\left(\sum_{\boldsymbol{y}\in S} e^{\boldsymbol{y}^T\boldsymbol{z}}\right)\right] .$$

But, since the second term is just a normalization constant, and hence independent of the random variable $\boldsymbol{x}$, we get that:

$$\bar{\ell}(\boldsymbol{z}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_S(\boldsymbol{z}^\star)} [-\boldsymbol{x}^T\boldsymbol{z}] + \log\left(\sum_{\boldsymbol{y}\in S} e^{\boldsymbol{y}^T\boldsymbol{z}}\right) .$$

Similarly, as in the proof of Lemma 3.5.1, one can compute the gradient with respect to $\boldsymbol{z}$ and get that:

$$\nabla_{\boldsymbol{z}}\bar{\ell}(\boldsymbol{z}) = -\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_S(\boldsymbol{z}^\star)}[\boldsymbol{x}] + \mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{D}_S(\boldsymbol{z})}[\boldsymbol{y}] \,. \tag{3.2}$$

Hence, computing in the exact same way the Hessian of $\bar{\ell}(\boldsymbol{z})$, we get the convexity of the negative log-likelihood for the population model with respect to the natural parameter vector $\boldsymbol{z}$.

Also, notice that the gradient $\nabla_{\boldsymbol{z}}\bar{\ell}(\boldsymbol{z})$ vanishes when $\boldsymbol{z} = \boldsymbol{z}^\star$. So, the true parameter vector $\boldsymbol{z}^\star$ minimizes the negative log-likelihood function of the truncated samples for the population model. This fact combined with the convexity of the population version of the negative log-likelihood yield the following.

Lemma 3.5.2. *For any $\boldsymbol{z}\in\mathbb{R}^d$, it holds that*

$$\bar{\ell}(\boldsymbol{z}^\star) \le \bar{\ell}(\boldsymbol{z}) \,,$$

where $\boldsymbol{z}^\star \in \mathbb{R}^d$ is the true parameter vector and $\bar{\ell}$ is the population negative log-likelihood objective, whose expectation is with respect to the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{z}^\star)$, for some arbitrary truncation set $S \subseteq \Pi_d$.

3.5.2 Initialization Lemma

Our next goal is to find a good initialization point for our SGD algorithm. Assume that for the truncation set S, it holds that $\mathcal{D}(\boldsymbol{p}^\star; S) = \alpha$. We claim that, if one draws $n = \widetilde{O}(d)$ samples $\{\boldsymbol{x}^{(t)}\}_{t=1}^n$ from the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$, the empirical mean

$$\widehat{\boldsymbol{p}}_S = \frac{1}{n}\sum_{t=1}^n \boldsymbol{x}^{(t)} \tag{3.3}$$

is close in L_2 distance to the true mean parameter vector $\boldsymbol{p}^\star$ with high probability.

In the following lemma, we provide the proximity result between the empirical mean $\widehat{\boldsymbol{p}}_S$ of the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$ and the true parameter vector $\boldsymbol{p}^\star$. This lemma will be useful in the upcoming section.

Lemma 3.5.3. *Let $\mathcal{D}(\boldsymbol{p}^\star)$ be the unknown Boolean product distribution and consider the truncation set $S \subseteq \Pi_d$ such that $\mathcal{D}(\boldsymbol{p}^\star; S) = \alpha$. The empirical mean $\widehat{\boldsymbol{p}}_S$, computed using $O\left(d \log(\frac{d}{\delta})\right)$ samples from the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$, satisfies:*

$$\|\widehat{\boldsymbol{p}}_S - \boldsymbol{p}^\star\|_2 \leq O\Big(\sqrt{\log(1/\alpha)}\Big)\,,$$

with probability $1-\delta$.

Proof. The proof of Lemma 3.5.3 can be decomposed in the following two lemmas. Combining the following two lemmas (we apply Lemma 3.5.4 with accuracy ϵ a small constant like $\sqrt{\log(1/\alpha)}/10$, since α is also a constant) using the triangle inequality for the L_2 norm, Lemma 3.5.3 follows. □

Lemma 3.5.4. *Consider $S \subseteq \Pi_d$ and let $\boldsymbol{p}_S$ be the parameter vector of the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$. There exists an algorithm that uses $O(\frac{d}{\epsilon^2}\log(\frac{d}{\delta}))$ samples from $\mathcal{D}_S(\boldsymbol{p}^\star)$ and computes an estimate $\widehat{\boldsymbol{p}}_S$ such that*

$$\|\widehat{\boldsymbol{p}}_S - \boldsymbol{p}_S\|_2 \leq \epsilon\,,$$

with probability $1-\delta$.

Proof. Consider the truncated true Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$ with truncation set $S \subseteq \Pi_d$. Consider the algorithm that, given n samples $\{\boldsymbol{x}^{(t)}\}$ from $\mathcal{D}_S(\boldsymbol{p}^\star)$, computes the empirical mean vector:

$$\widehat{\boldsymbol{p}}_S = \frac{1}{n}\sum_{t=1}^n \boldsymbol{x}^{(t)}\,.$$

Note that $\mathbb{E}\,\widehat{\boldsymbol{p}}_S = \boldsymbol{p}_S$. Fix a coordinate $j \in [d]$. By applying Hoeffding's inequality at $\widehat{p}_{S,j} = \frac{1}{n}\sum_{t=1}^n x_j^{(t)}$ (these random variables are bounded in $[0,1]$), one gets

$$\mathbf{Pr}\left[|\widehat{p}_{S,j} - p_{S,j}| > \epsilon/\sqrt{d}\right] \leq 2e^{-2n\frac{\epsilon^2}{d}}\,.$$

We can now use union bound and require the left hand side to be at most δ. Hence, we get that

$$2de^{-2n\frac{\epsilon^2}{d}} \leq \delta \Rightarrow n = \Omega\left(\frac{d}{\epsilon^2}\log\left(\frac{d}{\delta}\right)\right).$$

Consequently, given $\Theta(\frac{d}{\epsilon^2}\log(\frac{d}{\delta}))$ samples, we get that the empirical mean estimate $\boldsymbol{p}_S$ is within error ϵ in L_2 distance with probability $1-\delta$. □

Lemma 3.5.5. *Consider the unknown Boolean product distribution $\mathcal{D}(\boldsymbol{p}^\star)$ and a truncation set S such that $\mathcal{D}(\boldsymbol{p}^\star; S) = \alpha$. Let $\boldsymbol{p}_S$ be the parameter vector of the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$. Then, it holds that*

$$\|\boldsymbol{p}_S - \boldsymbol{p}^\star\|_2 \leq O\Big(\sqrt{\log(1/\alpha)}\Big).$$

Proof. Consider an arbitrary direction $\boldsymbol{w}$ with $\|\boldsymbol{w}\|_2 = 1$. Consider the random variable $\boldsymbol{w}^T\boldsymbol{x}$ where $\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{p}^\star)$. Note that $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}(\boldsymbol{p}^\star)}[\boldsymbol{w}^T\boldsymbol{x}] = \boldsymbol{w}^T\boldsymbol{p}^\star$. By applying Hoeffding's inequality:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}(\boldsymbol{p}^\star)}[\boldsymbol{w}^T\boldsymbol{x} > \boldsymbol{w}^T\boldsymbol{p}^\star + C] \leq e^{-2C^2}.$$

Hoeffding's inequality implies that the marginal of the true distribution in direction $\boldsymbol{w}$ has exponential tail and that holds for any (unit) direction. But, the worst case set S would assign mass α to the tail (in order to maximize the distance between the two means) and, hence:

$$\alpha \leq e^{-2C^2} \Rightarrow C = O\left(\sqrt{\log\frac{1}{\alpha}}\right).$$

The result follows.

□

3.5.3 Ball in the z-space

We will perform Projected SGD to a convex subspace of $\mathbb{R}^d$. The algorithm will optimize the negative log-likelihood for the population model $\overline{\ell}$ with respect to the natural parameters $\boldsymbol{z} = (z_1, \ldots, z_d)^T$ with $z_i = \log\frac{p_i}{1-p_i}$ in order to learn the true parameters $\boldsymbol{z}^\star = (z_1^\star, \ldots, z_d^\star)^T$ with $z_i^\star = \log\frac{p_i^\star}{1-p_i^\star}$. Our initial guess is $\widehat{\boldsymbol{z}} = (\widehat{z}_1, \ldots, \widehat{z}_d)^T$ with $\widehat{z}_i = \log\frac{\widehat{p}_{S,i}}{1-\widehat{p}_{S,i}}$. Afterwards, SGD will iterate over estimations $\boldsymbol{z}$ of the true parameters $\boldsymbol{z}^\star$.

In this section, we show that there exists a convex set that contains the true vector $\boldsymbol{z}^\star$ and each point in that set satisfies Assumptions 3 and 4.

In fact, we show that there exists a ball $\mathbb{B}$ of radius B centered at $\widehat{\boldsymbol{z}}$, that contains the true natural parameters $\boldsymbol{z}^\star$, with high probability. Additionally, every point $\boldsymbol{z}$ of that ball satisfies Assumptions 3 and 4. That is, for any $\boldsymbol{z} \in \mathbb{B}$, let $\mathcal{D}(\boldsymbol{z})$

be the Boolean product distribution and $\mathcal{D}_S(\boldsymbol{z})$ be an arbitrary truncation of $\mathcal{D}(\boldsymbol{z})$. Then, $\mathcal{D}_S(\boldsymbol{z})$ will be anti-concentrated too, in the sense of Assumption 3, and we will have $\mathcal{D}(\boldsymbol{z}; S) > c_\alpha$ for some constant c_a, that depends only on the initial mass of the set S. The existence of such a ball is presented in the following lemma.

Lemma 3.5.6. *There exists $B > 0$ such that the ball centered at the empirical estimate $\widehat{\boldsymbol{z}}$:*

$$\mathbb{B} = \{\boldsymbol{z} : \|\boldsymbol{z} - \widehat{\boldsymbol{z}}\|_2 \leq B\}$$

contains the true natural parameters, i.e.,

$$\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_2 \leq B\,,$$

with high probability, where the randomness is over the estimate $\widehat{\boldsymbol{z}}$.

Proof. We can assume that the real mean vector $\boldsymbol{p}^\star$ lies in $(0,1)^d$. Firstly, note that $\widehat{\boldsymbol{z}} \in (-\infty, \infty)^d$, since $(\widehat{\boldsymbol{z}})_i = \log \frac{\widehat{p}_{S,i}}{1-\widehat{p}_{S,i}}$ and $0 < \widehat{p}_{S,i} < 1$ for any $i \in [d]$. From now on, fix a coordinate $i \in [d]$ and consider the mapping $f(x) = \log \frac{x}{1-x}$ for $x \in (0,1)$. Note that f corresponds to the transformation of p_i to the natural parameter z_i and, hence:

$$|z_i^\star - \widehat{z}_i| = |f(p_i^\star) - f(\widehat{p}_{S,i})|\,.$$

Using the anti-concentration condition (see Assumption 3 and the discussion after this assumption), we get that there exists a positive constant γ such that $p_i^\star, \widehat{p}_{S,i} \in (\gamma, 1-\gamma)$ for any $i \in [d]$. Then, observe that there exists a positive finite constant C such f is C-Lipschitz in that interval. Hence,

$$|z_i^\star - \widehat{z}_i| = |f(p_i^\star) - f(\widehat{p}_{S,i})| \leq C|p_i^\star - \widehat{p}_{S,i}|\,.$$

Squaring each side and summing over $i \in [d]$, we get that

$$\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_2 \leq O\left(\sqrt{\log\frac{1}{\alpha}}\right)\,,$$

with high probability, where we used the proximity Lemma 3.5.3. Hence, the ball centered at $\widehat{\boldsymbol{z}}$ with radius $B = O\left(\sqrt{\log\frac{1}{\alpha}}\right)$, i.e., the set

$$\mathbb{B} = \{\boldsymbol{z} : \|\boldsymbol{z} - \widehat{\boldsymbol{z}}\|_2 \leq B\}$$

contains the true natural parameters $\boldsymbol{z}^\star$ and any point $\boldsymbol{z} \in \mathbb{B}$ is finite in each coordinate, since $\sum_{i=1}^d (z_i - \widehat{z}_i)^2 \leq B^2$. □

The value of B is equal to $O(\sqrt{\log(1/\alpha)})$. From now on, we will denote by $\mathbb{B}$ the ball of Lemma 3.5.6. In order to be able to perform the SGD algorithm, we have to prove that Assumptions 3 and 4 hold for any guess of our algorithm. Since the algorithm runs inside the ball $\mathbb{B}$, we have to prove that the two assumptions are preserved inside the ball. We remind the reader that any guess that lies outside the ball, is efficiently projected to its L_2 closest point $\boldsymbol{y} \in \mathbb{B}$.

Firstly, in Lemma 3.5.7, we prove that, in each iteration, every natural parameter vector $\boldsymbol{z}$ inside the ball $\mathbb{B}$, that corresponds to a mean vector $\boldsymbol{p}$ and induces a distribution $\mathcal{D}(\boldsymbol{p})$, will assign constant non-trivial mass to the set S.

Lemma 3.5.7 (Non-trivial mass inside the ball)**.** *Consider the true Boolean product distribution $\mathcal{D}(\boldsymbol{p}^\star)$ and $\mathcal{D}(\boldsymbol{p})$ be another Boolean product distribution such that the corresponding natural parameter vectors satisfy*

$$\|\boldsymbol{z}^\star - \boldsymbol{z}\|_2 \leq B = O\left(\sqrt{\log(1/\alpha)}\right).$$

Suppose that for a truncation set S we have that:

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{p}^\star)} [\mathbf{1}_{\boldsymbol{x} \in S}] \geq \alpha \,.$$

Then, it holds that

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{p})} [\mathbf{1}_{\boldsymbol{x} \in S}] \geq \mathrm{poly}(\alpha) \,.$$

Proof. Let $\mathcal{D}(\boldsymbol{p}^\star; S) = \alpha$ and $\mathcal{D}(\boldsymbol{p}; S) = \alpha'$. Firstly, notice that one can express the mass of the set S assigned by $\mathcal{D}(\boldsymbol{p})$ as:

$$\mathcal{D}(\boldsymbol{p}; S) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{p}^\star)} \left[\mathbf{1}_{\boldsymbol{x} \in S} \frac{\mathcal{D}(\boldsymbol{p}; \boldsymbol{x})}{\mathcal{D}(\boldsymbol{p}^\star; \boldsymbol{x})} \right].$$

This is equivalent to:

$$\mathcal{D}(\boldsymbol{p}; S) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{p}^\star)} \left[e^{-\log \frac{\mathcal{D}(\boldsymbol{p}^\star; \boldsymbol{x})}{\mathcal{D}(\boldsymbol{p}; \boldsymbol{x})}} \mathbf{1}_{\boldsymbol{x} \in S} \right].$$

We remind the reader that:

$$\mathcal{D}(\boldsymbol{z}; \boldsymbol{x}) = \exp(\boldsymbol{x}^T \boldsymbol{z}) \frac{1}{\prod_{i \in [d]} (1 + \exp(z_i))} \,.$$

Writing the log ratio in terms of the natural parameters $\boldsymbol{z}$, we get that:

$$\log \frac{\mathcal{D}(\boldsymbol{z}^\star; \boldsymbol{x})}{\mathcal{D}(\boldsymbol{z}; \boldsymbol{x})} = \boldsymbol{x}^T (\boldsymbol{z}^\star - \boldsymbol{z}) + C \,, \tag{3.4}$$

where $C = -\log \prod_{i \in [d]} (1 + e^{z_i^\star}) + \log \prod_{i \in [d]} (1 + e^{z_i}) = \log \frac{\prod_{i \in [d]} (1 - p_i^\star)}{\prod_{i \in [d]} (1 - p_i)}$ is independent of $\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{p}^\star)$. Since both $\boldsymbol{z}$ and $\boldsymbol{z}^\star$ lie inside the ball $\mathcal{B}$ and are finite, C corresponds to a constant. Now, set $g(\boldsymbol{x}) = \log \frac{\mathcal{D}(\boldsymbol{p}^\star; \boldsymbol{x})}{\mathcal{D}(\boldsymbol{p}; \boldsymbol{x})}$ and observe that:

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{p}^\star)} [g(\boldsymbol{x})] = D_{KL}(\mathcal{D}(\boldsymbol{p}^\star) \parallel \mathcal{D}(\boldsymbol{p})) \,.$$

Using Hoeffding's inequality on Equation (3.4), we get that:

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}(\boldsymbol{p}^\star)}\left[g(\boldsymbol{x}) - \mathbb{E}\, g \geq t\right] \leq \exp(-2t^2/\|\boldsymbol{z}^\star - \boldsymbol{z}\|_2^2)\,.$$

Setting $t = \sqrt{\log(2/\alpha)\|\boldsymbol{z}^\star - \boldsymbol{z}\|_2^2}$, it follows that:

$$\Pr_{\boldsymbol{x}\sim\mathcal{D}(\boldsymbol{p}^\star)}\left[g(\boldsymbol{x}) - \mathbb{E}\, g \geq \sqrt{\log(2/\alpha)\|\boldsymbol{z}^\star - \boldsymbol{z}\|_2^2}\right] \leq \alpha/2\,.$$

So, with probability at least $1-\alpha/2$, we get that the ratio $-g(\boldsymbol{x}) = -\log\frac{\mathcal{D}(\boldsymbol{p}^\star;\boldsymbol{x})}{\mathcal{D}(\boldsymbol{p};\boldsymbol{x})}$ will be at least

$$-\mathbb{E}\, g - \sqrt{\log(2/\alpha)\|\boldsymbol{z} - \boldsymbol{z}^\star\|_2^2}\,,$$

where we have that $\mathbb{E}\, g = D_{KL}(\mathcal{D}(\boldsymbol{p}^\star) \parallel \mathcal{D}(\boldsymbol{p})) \leq B^2$, by Proposition 3.1.3.($i$).

Hence, with probability at least $1-\alpha/2$, we get that the ratio $-\log\frac{\mathcal{D}(\boldsymbol{p}^\star;\boldsymbol{x})}{\mathcal{D}(\boldsymbol{p};\boldsymbol{x})}$ will be at least $-B^2 - B\sqrt{\log(2/\alpha)} = c\cdot\log(1/\alpha)$, for some constant c. Hence, $\alpha' \geq \frac{\alpha}{2}e^{-O(\log(1/\alpha))} = \mathrm{poly}(\alpha)$. This concludes the proof.

□

Applying the above lemma for the initial guess $\widehat{\boldsymbol{p}}_S$, we get that:

Corollary 3.5.8. *Consider a truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$ with mass $\mathcal{D}(\boldsymbol{p}^\star;S) \geq \alpha > 0$. The empirical mean $\widehat{\boldsymbol{p}}_S$, obtained by Lemma 3.5.3, satisfies $\mathcal{D}(\widehat{\boldsymbol{p}}_S;S) \geq c_\alpha$, with high probability, for some constant c_α that depends only on the constant $\alpha > 0$. The high probability result is over the randomness of the initialization $\widehat{\boldsymbol{p}}_S$.*

Hence, both at the initialization point $\widehat{\boldsymbol{z}}$ and while moving inside the ball $\mathbb{B}$ of Lemma 3.5.6, the mass assigned to the set S is always non-trivial.

We also need to show that the anti-concentration assumption is valid inside the ball $\mathcal{B}$. Assumption 3 states that the truncated distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$ of the true parameters is anti-concentrated. We will show that this holds for every truncated distribution $\mathcal{D}_S(\boldsymbol{z})$, induced by $\boldsymbol{z}$ that lies inside the ball $\mathbb{B}$. This is proven by the following lemma.

Lemma 3.5.9 (Anti-concentration inside the ball)**.** *Consider the true Boolean product distribution $\mathcal{D}(\boldsymbol{p}^\star)$ and $\mathcal{D}(\boldsymbol{p})$ be another Boolean product distribution such that the corresponding natural parameter vectors satisfy:*

$$\|\boldsymbol{z}^\star - \boldsymbol{z}\|_2 \leq B = O\left(\sqrt{\log(1/\alpha)}\right)\,.$$

Consider an arbitrary truncation set $S \subseteq \Pi_d$ such that $\mathcal{D}(\boldsymbol{p}^\star;S) \geq \alpha$. Assume that Assumption 3 holds for the true truncated distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$ with constant λ. Then, Assumption 3 still holds for $\mathcal{D}_S(\boldsymbol{p})$ with constant $\mathrm{poly}(\alpha,\lambda)$.

Proof. Consider the true Boolean product distribution $\mathcal{D}(\boldsymbol{p}^\star)$. Let S be the truncation set, where $\mathcal{D}(\boldsymbol{p}^\star; S) = \alpha$. The true truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$ satisfies Assumption 3. Hence, there exists a λ, such that, for any arbitrary hyperplane defined by $\boldsymbol{w} \in \mathbb{R}^d$ with $\|\boldsymbol{w}\|_2 = 1$ and $c \in \mathbb{R}$, we have that $\mathcal{D}_S(\boldsymbol{p}^\star; H) = \lambda$, where $H = \{\boldsymbol{x} : \boldsymbol{w}^T\boldsymbol{x} \notin (c-\lambda, c+\lambda)\} \subseteq \Pi_d$. Hence, the mass assigned by the true Boolean product distribution to the space $H \cap S$ is equal to $\mathcal{D}(\boldsymbol{p}^\star; H \cap S) = \lambda\alpha$.

Now, note that Lemma 3.5.7 holds for arbitrary set S. Hence, we can take the truncation set to be equal to $H \cap S$. Then, note that the hypotheses of Lemma 3.5.7 hold with $\mathcal{D}(\boldsymbol{p}^\star; H \cap S) \geq \lambda\alpha$. Applying the result of Lemma 3.5.7, we get that: $\mathcal{D}(\boldsymbol{p}; H \cap S) = \mathrm{poly}(\alpha, \lambda)$. Hence, $\mathcal{D}_S(\boldsymbol{p}; H) = \mathrm{poly}(\alpha, \lambda)$.

□

Applying the above lemma for the initial guess $\widehat{\boldsymbol{p}}_S$, we get that:

Corollary 3.5.10. *Consider a truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p}^\star)$ for which Assumption 3 holds. The truncated Boolean product distribution $\mathcal{D}_S(\widehat{\boldsymbol{p}}_S)$ induced by the empirical mean $\widehat{\boldsymbol{p}}_S$, obtained by Lemma 3.5.3, satisfies Assumption 3, with high probability over the randomness of the initialization $\widehat{\boldsymbol{p}}_S$.*

Hence, any natural parameter vector $\boldsymbol{z} \in \mathbb{B}$, induces a distribution $\mathcal{D}(\boldsymbol{z})$ such that the truncated distribution $\mathcal{D}_S(\boldsymbol{z})$ satisfies the anti-concentration assumption.

3.5.4 Unbiased Estimation of the Gradient

In this section, we discuss the rejection sampling algorithm in order to obtain an unbiased estimate for the gradient of the population version of the negative log-likelihood objective. Recall that

$$\nabla_{\boldsymbol{z}} \bar{\ell}(\boldsymbol{z}) = - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}_S(\boldsymbol{z}^\star)} [\boldsymbol{x}] + \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \mathcal{D}_S(\boldsymbol{z})} [\boldsymbol{y}] \,.$$

To compute an unbiased estimate for the first term, it suffices to draw a single sample from the distribution $\mathcal{D}_S(\boldsymbol{z}^\star)$ (we have oracle access to this distribution). For the second term, we perform rejection sampling as follows: we draw a vector $\boldsymbol{y} \sim \mathcal{D}(\boldsymbol{z})$ and we check whether $\boldsymbol{y} \in S$, using the membership oracle access to the truncation set S. If $\boldsymbol{y}$ lies in S, we use it to obtain the unbiased gradient estimate; otherwise, we reject this sample and repeat the procedure. We remind the reader that in each iteration we project the guess vector back to the feasible region $\mathbb{B}$. Since the mass of the set S inside the ball $\mathbb{B}$ is non-trivial and depends only on α (see Lemma 3.5.7), we get that the rejection sampling algorithm takes $\mathrm{poly}(1/\alpha)$ samples from the Boolean product distribution $\mathcal{D}(\boldsymbol{z})$ with high probability.

3.5.5 Strong-convexity of the negative log-likelihood

A crucial ingredient of our SGD algorithm is the strong convexity of $\overline{\ell}(\boldsymbol{z})$, that is the negative log-likelihood for the population model that corresponds to the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{z})$. Specifically:

Definition 3.5.11. *Let $f : \mathbb{R}^d \to \mathbb{R}$ with Hessian matrix $\boldsymbol{H}_f$. Then, f will be called λ-strongly convex if it holds that $\boldsymbol{H}_f \succeq \lambda \mathbb{I}$.*

As a last step before the analysis of our SGD algorithm, we will use Lemma 3.5.12 to show that $\overline{\ell}(\boldsymbol{z})$ is strongly convex for any $\boldsymbol{z} \in \mathbb{B}$. Let $\boldsymbol{H}_{\overline{\ell}}$ be the corresponding Hessian of $\overline{\ell}$ with the presence of arbitrary truncation $S \subseteq \Pi_d$.

Lemma 3.5.12 (Strong Convexity)**.** *Consider an arbitrary truncation set $S \subseteq \Pi_d$ whose mass with respect to the true Boolean product distribution is $\mathcal{D}(\boldsymbol{p}^\star; S) = \alpha$ and the truncated Boolean product distribution $\mathcal{D}_S(\boldsymbol{p})$ with the respective natural parameter $\boldsymbol{z}$ with $\boldsymbol{z} \in \mathbb{B}$. Then $\boldsymbol{H}_{\overline{\ell}}$ is $\lambda_{\boldsymbol{z}}$-strongly convex, where $\lambda_{\boldsymbol{z}} = \mathrm{poly}(\alpha, \lambda)$, where λ is introduced in Assumption 3.*

Proof. We have that $\boldsymbol{H}_{\overline{\ell}} = \mathrm{Cov}_{\boldsymbol{x} \sim \mathcal{D}_S(\boldsymbol{p})}[\boldsymbol{x}, \boldsymbol{x}]$. We will call this matrix $C_{\boldsymbol{p}}$. Then, we have that

$$C_{\boldsymbol{p}} = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}_S(\boldsymbol{p})} \left[(\boldsymbol{x} - \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \mathcal{D}_S(\boldsymbol{p})}[\boldsymbol{y}])(\boldsymbol{x} - \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \mathcal{D}_S(\boldsymbol{p})}[\boldsymbol{y}])^T \right].$$

For arbitrary vector $\boldsymbol{v} \in \mathbb{R}^d$ with $\|\boldsymbol{v}\|_2 = 1$, we have that to show that

$$\boldsymbol{v}^T C_{\boldsymbol{p}} \boldsymbol{v} > 0 .$$

Let us set $\boldsymbol{m} = \mathbb{E}_{\boldsymbol{y} \sim \mathcal{D}_S(\boldsymbol{p})}[\boldsymbol{y}]$. Note that

$$\boldsymbol{v}^T C_{\boldsymbol{p}} \boldsymbol{v} = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}_S(\boldsymbol{p})} [p_v(\boldsymbol{x})] ,$$

where, after some algebraic manipulation, we can get:

$$p_v(\boldsymbol{x}) = \sum_{j=1}^d v_j(x_j - m_j) \sum_{i=1}^d v_i(x_i - m_i) = (\boldsymbol{v}^T(\boldsymbol{x} - \boldsymbol{m}))^2 .$$

For the distribution $\mathcal{D}_S(\boldsymbol{p})$, Assumption 3 holds (using Lemma 3.5.9, since the respective natural parameters $\boldsymbol{z}$ lie inside the ball $\mathbb{B}$) with a positive constant $\lambda_{\boldsymbol{p}} = \mathrm{poly}(\alpha, \lambda)$. Specifically, setting $\boldsymbol{w} = \boldsymbol{v}$ and $c = \boldsymbol{v}^T \boldsymbol{m}$, Assumption 3 implies that there exists a positive constant $\lambda_{\boldsymbol{p}}$ such that:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{D}_S(\boldsymbol{p})} \left[|\boldsymbol{v}^T \boldsymbol{x} - c| > \lambda_{\boldsymbol{p}} \right] \geq \lambda_{\boldsymbol{p}} .$$

Hence, it follows that:

$$\boldsymbol{v}^T C_{\boldsymbol{p}} \boldsymbol{v} > \lambda_{\boldsymbol{p}}^3 > 0 ,$$

for any arbitrary unit vector $\boldsymbol{v} \in \mathbb{R}^d$. □

3.5.6 Analysis of SGD

Up to that point, we have showed that, using $\widetilde{O}\left(d\right)$ samples, there exists an initial guess, that is the empirical mean vector $\widehat{\boldsymbol{z}}$ such that there exists a ball $\mathbb{B}$ of radius $B = O\left(\sqrt{\log \frac{1}{\alpha}}\right)$ centered at the $\widehat{\boldsymbol{z}}$, that contains the true natural parameters $\boldsymbol{z}^\star$, with high probability. Additionally, every point that falls inside that ball satisfies Assumptions 3 and 4 and that $\overline{\ell}$ is strongly convex inside $\mathbb{B}$.

Apart from the previous analysis, in order to provide the theoretical guarantees of the Projected SGD algorithm, we have to show that, at each iteration, the square of the norm of the gradient vector of the $\overline{\ell}$ is bounded. This is proved in the following lemma.

Let $\boldsymbol{v}^{(t)}$ be the gradient of the negative log-likelihood that our SGD algorithm computes at step t. We remind the reader that $\boldsymbol{v}^{(t)} = -\boldsymbol{x}^{(t)} + \boldsymbol{y}$ (see Algorithm 4).

Lemma 3.5.13 (Bounded Variance Step)**.** *Let $\boldsymbol{z}^\star \in \mathbb{R}^d$ be the true natural parameter vector and let $\boldsymbol{z}$ be the guess after step $t-1$ according to which the gradient is computed. Assume that $\boldsymbol{z}$ and $\boldsymbol{z}^\star$ lie inside the ball $\mathbb{B}$ and that*

$$\min\{\mathcal{D}(\boldsymbol{z}; S), \mathcal{D}(\boldsymbol{z}^\star; S)\} \geq \beta\,.$$

Then, we have that:

$$\mathbb{E}\left[\|\boldsymbol{v}^{(t)}\|_2^2\right] \leq \frac{4d}{\beta}\,.$$

Proof. Let $\boldsymbol{p}$ (resp. $\boldsymbol{p}^\star$) be the corresponding mean parameter vector of the natural parameter vector $\boldsymbol{z}$ (resp. $\boldsymbol{z}^\star$). According to line 8 of the SGD Algorithm 4 and the Equation (3.2), we have that

$$\mathbb{E}\left[\|\boldsymbol{v}^{(t)}\|_2^2\right] = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_S(\boldsymbol{p}^\star)}\left[\mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{D}_S(\boldsymbol{p})}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\right]\,,$$

and hence

$$\mathbb{E}\left[\|\boldsymbol{v}^{(t)}\|_2^2\right] \leq 2\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_S(\boldsymbol{p}^\star)}\left[\|\boldsymbol{x}\|_2^2\right] + 2\mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{D}_S(\boldsymbol{p})}\left[\|\boldsymbol{y}\|_2^2\right]. \tag{3.5}$$

Now, since the measure of S is greater than β for both parameter vectors and since both parameters lie inside the ball, we can appropriately bound the above quantity. Observe that:

$$\mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{D}_S(\boldsymbol{p})}\left[\|\boldsymbol{y}\|_2^2\right] \leq \frac{1}{\beta}\mathop{\mathbb{E}}_{\boldsymbol{y}\sim\mathcal{D}(\boldsymbol{p})}\left[\|\boldsymbol{y}\|_2^2\right] \leq \frac{d}{\beta}\,.$$

Similarly, we have that:

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}_S(\boldsymbol{p}^\star)}\left[\|\boldsymbol{x}\|_2^2\right] \leq \frac{d}{\beta}\,.$$

The result follows by combining the two inequalities to Equation (3.5). □

Let $\bar{\ell}$ be the negative log-likelihood for the population model. We present a folklore SGD theorem. The formulation we use is from (SSBD14).

Fact 1. *Let $f = \bar{\ell}$. Assume that f is μ-strongly convex, that $\mathbb{E}[\boldsymbol{v}^{(t)}|\mathbf{z}^{(t-1)}] \in \partial f(\mathbf{z}^{(t-1)})$ and that $\mathbb{E}\left[\|\boldsymbol{v}^{(t)}\|_2^2\right] \leq \rho^2$. Let $\mathbf{z}^\star \in \operatorname{argmin}_{z\in\mathcal{B}} f(\mathbf{z})$ be an optimal solution. Then,*

$$\mathbb{E}[f(\overline{\mathbf{z}})] - f(\mathbf{z}^\star) \leq \frac{\rho^2}{2\mu M} \cdot (1 + \log M)\,,$$

where $\overline{\mathbf{z}}$ is the output of the SGD Algorithm 4.

As an application of Fact 1 and Lemma 3.5.13, we obtain directly the following result.

Lemma 3.5.14. *Let $\boldsymbol{z}^\star$ be the true parameters of the model, let $f = \bar{\ell}$ be defined as above, $\beta = \min_{\boldsymbol{z}\in\mathcal{B}} D(\boldsymbol{z}; S), \mu \geq \min_{\boldsymbol{z}\in\mathcal{B}} \lambda_{\boldsymbol{z}}$, then there exists a universal constant $C > 0$ such that*

$$\mathbb{E}[f(\overline{\boldsymbol{z}})] - f(\boldsymbol{z}^\star) \leq \frac{Cd}{\beta\mu M} \cdot (1 + \log M)\,.$$

We are now ready to prove our main Theorem 3.4.1.

Proof. Using Lemma 3.5.14 and applying Markov's inequality, it follows that:

$$\mathbf{Pr}\left[f(\overline{\boldsymbol{z}}) - f(\boldsymbol{z}^\star) \geq \frac{3Cd}{\beta\mu M} \cdot (1 + \log M)\right] \leq \frac{1}{3}\,.$$

We can amplify the probability of success to $1 - \delta$ by repeating $N = \log(1/\delta)$ independently from scratch the SGD procedure and keeping the estimation that achieves the maximum log-likelihood value. The procedure is completely similar to the proof of Theorem 1 of (DGTZ18) and we repeat it here for completeness. Let $\mathcal{E}$ be the set of our N estimates. The optimal estimate would be $\widetilde{\boldsymbol{z}} = \operatorname{argmin}_{\boldsymbol{z}\in\mathcal{E}} \bar{\ell}(\boldsymbol{z})$, but we cannot compute exactly $f = \bar{\ell}$. Using the Markov's inequality, we get that, with probability at least $1 - \delta$, at least $2/3$ of our estimates satisfy

$$f(\boldsymbol{z}) - f(\boldsymbol{z}^\star) \leq \frac{3Cd}{\beta\mu M} \cdot (1 + \log M)\,.$$

Let us set $\zeta := \frac{3Cd}{\beta\mu M}(1+\log M)$. As we will see, using the strong convexity property, we get that $f(\boldsymbol{z}) - f(\boldsymbol{z}^\star)$, implies $\|\boldsymbol{z} - \boldsymbol{z}^\star\|_2 \leq c\zeta$, for some c. Hence, with high probability $1-\delta$ for at least $2/3$ of our estimations, the L_2 norm is at most $2c\zeta$. So, we can set appropriately the value of $\widetilde{\boldsymbol{z}}$ to be a point that is at least $2c\zeta$ close to more that the half of our N estimations. That value will satisfy $f(\widetilde{\boldsymbol{z}}) - f(\boldsymbol{z}^\star) \leq \zeta$. Now, using Lemmata 3.5.9 and 3.5.7, there are quantities $c_\alpha = \mathrm{poly}(\alpha), c_{\alpha,\lambda} = \mathrm{poly}(\alpha, \lambda)$ such that $\beta \geq c_\alpha$ and $\mu \geq c_{\alpha,\lambda}$, where α is the constant of Assumption 4 and λ is the parameter of Assumption 3. This leads to the following statement:

With probability at least $1 - \delta$, we have that: $f(\widetilde{\boldsymbol{z}}) - f(\boldsymbol{z}^\star) \leq c' \frac{d}{M}(1 + \log M)$, where c' is $\mathrm{poly}(1/\alpha, 1/\lambda)$. Now, we can use the Lemma 13.5 of (SSBD14) about strong convexity:

Fact 2. *If f is μ-strongly convex and $\boldsymbol{z}^\star$ is a minimizer of f, then, for any $\boldsymbol{z}$, it holds that:*

$$f(\boldsymbol{z}) - f(\boldsymbol{z}^\star) \geq \frac{\mu}{2}\|\boldsymbol{z} - \boldsymbol{z}^\star\|_2^2 \,.$$

Using this result, we can get

$$\|\widetilde{\boldsymbol{z}} - \boldsymbol{z}^\star\|_2 \leq c''\sqrt{\frac{d}{M}\cdot(1+\log M)}\,,$$

where c'' is $\mathrm{poly}(1/\alpha, 1/\lambda)$.

Hence, the number of samples is $O(NM)$ and the running time is $\mathrm{poly}(N, M, d, 1/\epsilon)$. For $N = \log(1/\delta)$ and $M \geq \mathrm{poly}(1/\alpha, 1/\lambda)\widetilde{O}\left(\frac{d}{\epsilon^2}\right)$, the result follows. □

3.6 Appendix: Deferred Proofs

In this section, we provide the proof of Proposition 3.1.2.

Proof. We define the pair of functions on the space $(p, q) \in (0,1)^2$:

$$f(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$$

and

$$g(p,q) = \left(\log\frac{p}{1-p} - \log\frac{q}{1-q}\right)^2 .$$

Both functions have a root at $p = q = 1/2$. Notice that g is symmetric. Fix q. We will denote with f_q (resp. g_q) the projection of f (resp. g) in the p-space, having fixed q. Then, $f_q(q) = g_q(q) = 0$ is the unique root for $p \in (0,1)$. Let $h(p) = f_q(p) - g_q(p)$. We claim that h has a unique root at q for $p \in (0,1)$. The derivate of h with respect to p is equal to:

$$\frac{dh}{dp} = \log\left(\frac{p(1-q)}{q(1-p)}\right)\left(1 - \frac{2}{p(1-p)}\right).$$

Notice that: $1 - \frac{2}{p(1-p)} < 0 \;\forall p \in (0,1)$ and that:

$$\log\left(\frac{p(1-q)}{q(1-p)}\right) = \begin{cases} < 0 & \text{for } p < q\,, \\ 0 & \text{for } p = q\,, \\ > 0 & \text{for } p > q\,. \end{cases}$$

Hence, $h'(q) = 0$ and, hence, h is strictly increasing for $p < q$ and h is strictly decreasing for $p > q$. Also, $p = q$ is the unique solution of the equation $h(p) = 0$ for $p \in (0,1)$.

For $p < q \Rightarrow h(p) < 0 \Rightarrow f_q(p) < g_q(p)$ and for $p > q \Rightarrow h(p) < 0 \Rightarrow f_q(p) < g_q(p)$. So, the desired inequality holds for the arbitrary fixed $q \in (0,1)$. Hence, the inequality follows for every $p, q \in (0,1)$. □

Chapter 4

Learning from Coarse Labels

We decompose this chapter into two parts. The first one deals with classification with coarse labels and the second one with the problem of estimating the mean of a Gaussian distribution from coarse data.

4.1 Classification with Coarse Data

We begin this chapter by reminding to the reader the generative model of coarsely labeled data that we consider in this semi-supervised setting. We model coarse labels as subsets of the domain of all possible fine labels. Our coarse data model is defined as follows.

Definition 4.1.1 (Generative Process of Coarse Data with Context)**.** *Let $\mathcal{X}$ be an arbitrary domain, and let $\mathcal{Z} = \{1, \dots, k\}$ be the discrete domain of all possible fine labels. We generate coarsely labeled examples as follows: (i) Draw a finely labeled example (x, z) from a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Z}$, (ii) Draw a coarsening partition $\mathcal{S}$ (of $\mathcal{Z}$) from a distribution π, (iii) Find the unique set $S \in \mathcal{S}$ that contains the fine label z. Finally, we observe the coarsely labeled example (x, S). We denote $\mathcal{D}_\pi$ the distribution of the coarsely labeled example (x, S).*

In this chapter, our main focus is to answer the following question which summarizes the challenges behind the coarse labels' problem.

Question 2. *Can we train a model, using coarsely labeled examples $(x, S) \sim \mathcal{D}_\pi$, that classifies finely labeled examples $(x, z) \sim \mathcal{D}$ with accuracy comparable to that of a classifier that was trained on examples with fine grained labels?*

The above Definition 4.1.1 imposes no restrictions on the distribution over partitions π. In order for Question 2 to be statistically possible, we need to consider distributions over partitions π that preserve fine-label information. The following definition quantifies this by stating that reasonable partition distributions π are those that preserve the total variation distance between different distributions supported on the domain of the fine labels $\mathcal{Z}$. We remark that the following definition does not require $\mathcal{D}$ to be supported on pairs (x, z) but is a general statement for the unsupervised version of the problem, see also Definition 4.1.4.

Definition 4.1.2 (Information Preserving Partition Distribution). *Let $\mathcal{Z}$ be any domain and let $\alpha \in (0, 1]$. We say that π is an α-information preserving partition distribution if for every two distributions $\mathcal{D}^1, \mathcal{D}^2$ supported on $\mathcal{Z}$, it holds that $d_{\mathrm{TV}}(\mathcal{D}^1_\pi, \mathcal{D}^2_\pi) \geq \alpha \cdot d_{\mathrm{TV}}(\mathcal{D}^1, \mathcal{D}^2)$, where $d_{\mathrm{TV}}(\mathcal{D}^1, \mathcal{D}^2)$ is the total variation distance of $\mathcal{D}^1$ and $\mathcal{D}^2$.*

Crucially, our generative model allows the partitions to have arbitrarily complex combinatorial structure that makes the process of “inverting" the partition transformation computationally challenging.

We continue with the main result of this chapter. Formally, we design an algorithm (Algorithm 5) that, given coarsely labeled examples (x, S), efficiently simulates statistical queries over finely labeled examples (x, z). Surprisingly, the runtime and sample complexity of our algorithm do not depend on the combinatorial structure of the partitions, but only on the number of fine labels k and the information preserving parameter α of the partition distribution π.

Theorem 4.1.3 (SQ from Coarsely Labeled Examples). *Consider a distribution $\mathcal{D}_\pi$ over coarsely labeled examples in $\mathbb{R}^d \times [k]$, (see Definition 4.1.1) with α-information preserving partition distribution π. Let $q : \mathbb{R}^d \times [k] \to [-1, 1]$ be a query function, that can be evaluated on any input in time T, and $\tau, \delta \in (0, 1)$. There exists an algorithm (Algorithm 5), that draws $N = \widetilde{O}(k^4/(\tau^3\alpha^2)\log(1/\delta))$ coarsely labeled examples from $\mathcal{D}_\pi$ and, in $\mathrm{poly}(N, T)$ time, computes an estimate $\widehat{r}$ such that, with probability at least $1 - \delta$, it holds $\left|\mathbb{E}_{(x,z)\sim\mathcal{D}}[q(x, z)] - \widehat{r}\right| \leq \tau$.*

Before proving the above result, we review the prior work on the field.

4.1.1 Related Work

Our work is closely related to the literature of learning from censored-truncated data and learning with noise. There has been a large number of recent works dealing inference with truncated data from a Gaussian distribution (DGTZ18; KTZ19), mixtures of Gaussians (NP19), linear regression (DGTZ19b; IZD20; DRZ20), sparse Graphical models (BDNP20) or Boolean product distributions (FKT20), and nonparametric estimation (DKTZ21). A significant feature of our work is that it can capture the closely related field of censored statistics (Coh16; Bre96; Wol79).

The area of robust statistics (Hub04) is also very related to our work as it also deals with biased data-sets and aims to identify the distribution that generated the data. Recently, there has been a large volume of theoretical work for computationally-efficient robust estimation of high-dimensional distributions (DKK$^+$16;

CSV17; LRV16b; DKK$^+$17; DKK$^+$18; KKM18; HL19; DKK$^+$19a; CDGS20; BDH$^+$20) in the presence of arbitrary corruptions to a small ε fraction of the samples.

The line of research dealing with statistical queries (Kea98; BFKV98; FPV15; FGV17; Fel17; FGR$^+$17; DKS17b) is closely related to one of our main results (Theorem 4.1.3). It is generally believed that SQ algorithms capture all reasonable machine learning algorithms (AD98; BFKV98; BDMN05; DV08; FGR$^+$17; BF15; FGV17) and there is a rich line of research indicating SQ lower-bounds for these classes of algorithms (FGR$^+$17; DKS17b; Sha18; VW19**?** ; DKZ20; GGJ$^+$20; GGK20).

Another strand of research closely related to our problem is the Partial Label Learning task: a weakly supervised learning problem where each training example is associated with a set of candidate labels among which only one is true (CST11b; CPCP14; YZ16a). For a small sample of the numerous works related to this problem from the applied CS community, we refer the reader to (NC08a; JG02; ZY15; ZZL16; ZYT17a; XLG19; XQGZ21; WZL21) and the references therein.

The problem of learning from coarse labels falls in the regime of semi-supervised learning (CSZ06) and it appears in various literatures (at least) termed as (i) partial label learning (CST11a), (ii) ambiguous label learning (CSJT09; HB06), (iii) superset label learning (HC15) and (iv) soft label learning (CODA08). Closely related to these tasks are the problems of learning from complementary labels (INHS17) and, more generally, learning from noisy and corrupted examples (AL88; SBH13; BS14; VRW17; LBMK20).

We stick with the term partial label learning for now since this is the most widely used. Many real-world learning tasks were solved under the framework of partial label learning such as multimedia content analysis (CSJT09; CST11a) and semantic image segmentation (PCMY15). For instance, in the breakthrough work of (CSJT09), one of the experiments conducted goes as follows: Using a screenplay (a specific shot from a movie and the associated dialogues), the observer (i.e., the learning algorithm) can tell who appears in a given scene (from the names participating in the dialogue), but for each face detected in the scene, the person's identity is ambiguous. Hence, each face is partially labeled with the set of characters appearing in the scene (RBK07). The goal is to remove this ambiguity and learn the name of each character from few screenplays.

We refer to (JG02; NC08b) and the references therein for some seminal papers in the area. Through the years, various approaches have been proposed to solve this challenging and interesting problem by utilizing major machine learning techniques, such as maximum likelihood estimation and Expectation-Maximization (JG02), convex optimization (CST11a), k-nearest neighbors (HB06) and error-correcting output codes (Zha14; ZYT17b). For an overview of the practical treatment on the problem, we refer the interested reader to (YZ16b; XQGZ21; WCH$^+$21) (and the references therein) and more broadly to (TGH15; VEH20).

Despite extensive studies on partial label learning from an industrial perspective (applied ML), our theoretical level of understanding is still limited. A fundamental line of research deals with the statistical consistency (see e.g., (CST11a; CSGGSR14; FLH$^+$20; CRB20; LXF$^+$20; WCH$^+$21)) and the learnability (LD14) of partial label learning algorithms. Moreover, (CGAD22) present a methodology between partial supervision and validation, developing a conformal prediction framework (SV08).

Closer to our learning from coarse labels approach are the works of (CS12) and (VRW17). In the former, the goal is to estimate the posterior class probabilities from partially labelled data while, in the latter, the authors study a more general problem of learning from corrupted labels and aim to "invert" the corruption. Their technique is inspired by the work of (NDRT13), where they proposed the method of unbiased estimators (which is quite close to the connection between random classification noise and the SQ framework of (Kea98)). This backward correction procedure of (NDRT13; CS12; VRW17) recovers the information lost from the corrupted labels (under some structural assumptions) and results in an unbiased estimate of the risk with respect to true distribution. Crucially, these works have to assume that the corruption process (i.e., the coarsening mechanism) is known. This is also commented in (CRB20). Our SQ reduction does not require to know the mechanism; in some sense, the algorithm uses rejection sampling and learning coarse discrete distributions (which is an unsupervised learning problem) in order to invert the coarsening in the sense of (VRW17) and obtain statistical queries with respect to the distribution over the finely-labeled examples.

4.1.2 Technical Details for Learning from Coarse Labels

In this section, we consider the problem of *supervised* learning from coarse data. In this setting, there exists some underlying distribution over finely labeled examples, $\mathcal{D}$. However, we have sample access only to the distribution associated with coarsely labeled examples $\mathcal{D}_\pi$, see Definition 4.1.1. Under this setting, even problems that are naturally convex when we have access to examples with fine labels, become non-convex when we introduce coarse labels (e.g., multiclass logistic regression). The main result of this section is Theorem 4.1.3, which allows us to compute statistical queries over finely labeled examples.

Overview of the Proof of Theorem 4.1.3

In order to simulate a statistical query we take a two step approach. Our first building block considers the unsupervised version of the problem, see Definition 4.1.4, i.e., we marginalize the context x and try to learn the distribution of the fine labels z given coarse samples S. This can be viewed as learning a general discrete distribution supported on $\mathcal{Z} = \{1, \dots, k\}$ given coarse samples, i.e., subsets of $\mathcal{Z}$. We show that, when the partition distribution π is α-information preserving, this can be done efficiently, see Proposition 4.1.5. Our algorithm (Algorithm 5) exploits the fact that even though in general having coarse data results in non-concave likelihood objectives, when we consider parametric models (see, for example, the case of logistic regression in Section 4.1.4), this is not true when we maximize over all discrete distributions. In Proposition 4.1.5, we show that $\widetilde{O}(k/(\epsilon\alpha)^2)$ samples are sufficient for this step. For the details of this step, see Section 4.1.2.

Using the above algorithm, one could try to separately learn the marginal distribution over x, $\mathcal{D}_x$ and the distribution of the fine labels z *conditional on some fixed* x; let us denote this distribution as $\mathcal{D}_z^x$. Then one could generate finely labeled examples (x, z) and use them to estimate the query $\mathbb{E}_{(x,z)\sim\mathcal{D}}[q(x, z)]$. The reason that this naive approach fails is that it requires many coarse examples (x, S) with exactly the same value of x. Unless the domain $\mathcal{X}$ is very small, the probability that we observe samples with the same value of x is going to be tiny. In order to overcome this obstacle, at a high level, our approach is to split the domain $\mathcal{X}$ into larger sets and then, learn the conditional distribution of the labels, not on a fixed point x, but on these larger sets of non-trivial mass.

Intuitively, in order to have an effective partition of the domain $\mathcal{X}$, we want to group together points x whose values $q(x, z)$ are close. Since z belongs in a discrete domain $\mathcal{Z} = [k]$, we can decompose the query $q(x, z)$ as $q(x, z) = \sum_{i=1}^{k} q(x, i)\mathbf{1}\{z = i\}$. We estimate the value of $\mathbb{E}_{(x,z)\sim\mathcal{D}}[q(x, i)\mathbf{1}\{z = i\}]$ separately. To find a suitable reweighting of the domain $\mathcal{X}$, we perform rejection sampling, accepting a pair $(x, S) \sim \mathcal{D}$ with probability $q(x, i)$ [1]: points x that have small value $q(x, i)$ contribute less in the expectation and are less likely to be sampled. After performing this rejection sampling process based on x, we have pairs (x, S), conditional that x was accepted. Now, using our previous maximum likelihood learner of Proposition 4.1.5 we learn the marginal distribution over fine labels and use it to answer the query. We provide the details of this rejection sampling step in the full proof of Theorem 4.1.3, see Section 4.1.2.

For a description of the corresponding algorithm that simulates statistical queries, see Algorithm 5. To keep the presentation simple we state the algorithm for the case where the query function $q(x, z)$ is positive. It is straightforward to generalize it for general queries, see Section 4.1.2.

Remark 1 (Empirical Likelihood Approach)**.** *One could try to use the empirical likelihood directly over the coarsely labeled data (as defined in (Owe01)). However, in general, these empirical likelihood objectives are non-convex when the data are coarse and therefore it is computationally hard to optimize them directly. Our approach for simulating statistical queries consists of two ingredients: reweighting the feature space via rejection sampling in order to group together points and learning discrete distributions from coarse data. To learn the discrete distributions (Section 4.1.2), we use a (direct) empirical likelihood approach similar to that of (Owe88; Owe90; Owe01). However, our main contribution is the use of rejection sampling to reduce the initial non-convex problem to the special case of learning a*

[1] It is easy to handle the case where this function takes negative values, see the proof of Theorem 4.1.3.

Algorithm 5 Statistical Queries from Coarse Labels.

```
Input: Query $q : \mathcal{X} \times \mathcal{Z} \mapsto (0,1]$, tolerance $\tau \in [0,1]$, confidence $\delta \in [0,1]$.
Oracle: Access to coarsely labeled samples $(x, S) \sim \mathcal{D}_\pi$, $\pi$ is $\alpha$-information preserving.
Output: Estimate $\widehat{r}$ such that $\left|\mathbb{E}_{(x,z)\sim\mathcal{D}}[q(x,z)] - \widehat{r}\right| \leq \tau$ with probability at least $1-\delta$.

procedure StatQuery$(q, \tau, \delta)$
    Compute $\widehat{r}_i \leftarrow \mathrm{SQ}(q, i, O(\tau/k), \delta/k)$ for any $i \in \mathcal{Z}$.
    Output $\widehat{r} \leftarrow \sum_{i=1}^{k} \widehat{r}_i$.

procedure SQ$(q, i, \rho, \delta)$
    Draw $N_1 = \widetilde{\Theta}\big(\frac{\log(1/\delta)}{\rho^2}\big)$ samples $(x_j, S_j)$ from $\mathcal{D}_\pi$.
    Compute $\widehat{\mu}_i \leftarrow \frac{1}{N_1}\sum_{j=1}^{N_1} q(x_j, i)$.
    if $\widehat{\mu}_i \leq \rho$ do
        Output $\widehat{r}_i \leftarrow 0$.
    end
    Draw $N_2 = \widetilde{\Theta}\big(\frac{k \log(1/\delta)}{\rho^3\alpha^2}\big)$ samples $(x_j, S_j)$ from $\mathcal{D}_\pi$.    ▷ $\widetilde{\Theta}\big(\frac{k^4 \log(1/\delta)}{\tau^3\alpha^2}\big)$ examples overall.
    $T_{accept} \leftarrow \emptyset$.    ▷ Training set of accepted samples.
    Add $S_j$ in $T_{accept}$ with probability $q(x_j, i)$, $\forall j \in [N_2]$.    ▷ Rejection Sampling Process.
    Compute $\widetilde{\mathcal{D}}$ using Proposition 4.1.5 with input $(T_{accept}, \rho, \delta)$.
    Output $\widehat{r}_i \leftarrow \widehat{\mu}_i \cdot \widetilde{\mathcal{D}}(i)$.
```

discrete distribution (with small support) from coarse data which, as we prove, is a tractable (convex) problem. For more connections with censored statistics techniques, we refer the reader to (TG75; Owe88; GVDLR97; Owe01).

Learning Marginals Over Fine Labels

In this subsection, we deal with *unsupervised* learning from coarse data in discrete domains. Although this is an ingredient of our main result for simulating statistical queries in a supervised setting where labeled data (x, S) are given, the result of this section does not depend on the points x and concerns the unsupervised version of the problem. To keep the notation simple, we will use $\mathcal{D}$ to denote a distribution over finite labels $\mathcal{Z}$.

Definition 4.1.4 (Generative Process of Coarse Data)**.** *Let $\mathcal{Z}$ be a discrete domain and $\mathcal{D}$ be a distribution supported on $\mathcal{Z}$. Moreover, let π be a distribution supported on partitions of $\mathcal{Z}$. We consider the following generative process:*

1. *Draw z from $\mathcal{D}$.*
2. *Draw a partition $\mathcal{S}$ from the distribution over all partitions π.*
3. *Observe the set $S \in \mathcal{S}$ that contains z.*

We denote the distribution of S as $\mathcal{D}_\pi$.

The assumption that we require is that the partition distribution π is α-information preserving, see Definition 4.1.2. At this point we give some examples of information preserving partition distributions. We first observe that $\alpha = 0$ if and only if the problem is not identifiable. For instance, if π is supported only on the partition $\mathcal{S} = \{\{1, 2\}, \{3, \ldots, k\}\}$, the problem is not identifiable, since, for example, the fine label 1 is indistinguishable from the fine label 2. The value $\alpha = 1$ is attained when the partition totally preserves the distribution distance. Intuitively, the value $1 - \alpha$ corresponds to the distortion that the coarse labeling introduces to a finely labeled dataset.

In many cases most fine labels may be missing. Consider two data providers that use different methods to round their samples. The rounding's uncertainty can be viewed as a coarse labeling of the data. Assume that we add discrete (balanced Bernoulli) noise ξ to some true value $x \in [0..k]$. Consider two partitions $\{\mathcal{S}_1, \mathcal{S}_2\}$ with $\mathcal{S}_1 = \{\{0, 1\}, \{2, 3\}, \ldots, \{k-1, k\}, \{k+1\}\}$ and $\mathcal{S}_2 = \{\{0\}, \{1, 2\}, \ldots \{k-1, k\}\}$. Observe that, when $x + \xi$ is odd, we can think of the rounded sample, as a draw from $\mathcal{S}_1$ and when $x + \xi$ is even, as a draw from $\mathcal{S}_2$. This example shows that we can capture the problem of deconvolution of two distributions $\mathcal{D}_1, \mathcal{D}_2$, where one of them is known and we observe samples $x_1 + x_2, x_i \sim \mathcal{D}_i$.

The following proposition establishes the sample complexity of unsupervised learning of discrete distributions with coarse data. Our goal is to compute an estimate of the discrete distribution $\mathcal{D}^\star$ with probability vector $\boldsymbol{p}^\star \in \Delta^k$ from N coarse samples $S_1, \ldots, S_N$ drawn from the distribution $\mathcal{D}^\star_\pi$. Our algorithm maximizes the empirical likelihood. Analyzing the empirical log-likelihood objective $\mathcal{L}_N(\boldsymbol{p}) = \frac{1}{N} \sum_{n=1}^N \log\left(\sum_{i \in S_n} \boldsymbol{p}_i\right)$, where $\boldsymbol{p} \in \Delta^k$ is a guess probability vector, we observe that the problem is concave and, therefore, can be efficiently optimized (e.g., by gradient descent).

Proposition 4.1.5. *Let $\mathcal{Z}$ be a discrete domain of cardinality k and let $\mathcal{D}$ be a distribution supported on $\mathcal{Z}$. Moreover, let π be an α-information preserving partition distribution for some $\alpha \in (0, 1]$. Then, with $N = \widetilde{O}(k/(\epsilon^2\alpha^2)\log(1/\delta))$ samples from $\mathcal{D}_\pi$ and in time polynomial in the number of samples N, we can compute a distribution $\widetilde{\mathcal{D}}$ supported on $\mathcal{Z}$ such that $d_{\mathrm{TV}}(\widetilde{\mathcal{D}}, \mathcal{D}) \le \epsilon$.*

Proof. Let $\mathcal{D}^\star$ be the target discrete distribution, supported on a discrete domain of size k, and let $\boldsymbol{p}^\star \in \Delta^k$ be the corresponding probability vector. For some distribution $\mathcal{D}$ supported on a discrete domain of size k, we define the following population log-likelihood objective.

$$\mathcal{L}(\mathcal{D}) = \mathop{\mathbb{E}}_{S \sim \mathcal{D}^\star_\pi}[\log \mathcal{D}(S)] = \mathop{\mathbb{E}}_{S \sim \mathcal{D}^\star_\pi}\left[\log\Big(\sum_{i \in S} \mathcal{D}(i)\Big)\right]. \tag{4.1}$$

Since $\mathcal{D}$ is a discrete distribution for simplicity we may identify with its probability vector $\boldsymbol{p}$, where $\boldsymbol{p}_i = \mathcal{D}(i)$. Therefore, for any $\boldsymbol{p}$ in the probability simplex Δ^k, we define

$$\mathcal{L}(\boldsymbol{p}) = \mathop{\mathbb{E}}_{S \sim \mathcal{D}_\pi^\star} \left[\log \sum_{i \in S} \boldsymbol{p}_i \right] . \tag{4.2}$$

The corresponding empirical log-likelihood objective after drawing N independent samples $S_1, \ldots, S_N$ from $\mathcal{D}_\pi^\star$ is given by

$$\mathcal{L}_N(\boldsymbol{p}) = \frac{1}{N} \sum_{n=1}^{N} \log \left(\sum_{i \in S_n} \boldsymbol{p}_i \right) . \tag{4.3}$$

We first observe that the log-likelihood (both the population and the empirical) is a concave function and therefore can be efficiently optimized (e.g., by gradient descent). Thus, our main focus in this proof is to bound its sample complexity. We first observe that when the guess $\boldsymbol{p} \in \Delta^k$ has some very biased coordinates, i.e., for some subset S the corresponding $\boldsymbol{p}_i$'s are close to 0, the probability of a set S, $\sum_{i \in S} p_i$ will be close to zero and therefore $\log \left(\sum_{i \in S} \boldsymbol{p}_i \right)$ will be large. Thus, we have to restrict our search to a subset of the probability simplex, i.e., have $\boldsymbol{p}_i \geq \epsilon/k$. We set $\widetilde{\Delta}^k = \{ \boldsymbol{p} \in \Delta^k, \boldsymbol{p}_i \geq \epsilon/k \text{ for all } i = 1, \ldots, k \ \}$. We now prove that, given roughly $k/(\epsilon^2 \alpha^2)$ samples, we can guarantee that probability vectors that are far from the optimal vector $\boldsymbol{p}^\star$ will also be significantly sub-optimal in the sense that they are far from being maximizers of the empirical log-likelihood.

Claim 1. *Let $N \geq \widetilde{\Omega}(k/(\epsilon^2\alpha^2) \log(1/\delta))$. With probability at least $1 - \delta$, we have that, for every $\boldsymbol{p} \in \widetilde{\Delta}^k$ such that $\|\boldsymbol{p} - \boldsymbol{p}^\star\|_1 \geq \epsilon$, it holds*

$$\max_{\boldsymbol{q} \in \widetilde{\Delta}^k} \mathcal{L}_N(\boldsymbol{q}) - \mathcal{L}_N(\boldsymbol{p}) \geq \Omega\left((\epsilon\alpha)^2 \right) .$$

Proof. We first construct a cover of the probability simplex $\widetilde{\Delta}^k$ by discretizing each coordinate $\boldsymbol{p}_i$ to integer multiples of $O((\epsilon^{3/2}\alpha/k)^2)$. The resulting cover $\mathcal{C}$ contains $O((k/(\epsilon^{3/2}\alpha))^{2k})$ elements. We first observe that we can replace any element $\boldsymbol{p} \in \widetilde{\Delta}^k$ with an element $\boldsymbol{p}'$ inside our cover $\mathcal{C}$ without affecting the value of the objective $\mathcal{L}_N(\boldsymbol{p})$ by a lot. In particular, using the fact that $x \mapsto \log(x)$ is $1/r$-Lipschitz in the interval $[r, +\infty)$, we have that for any set $S \subseteq \{1, \ldots, k\}$ it holds

$$\left| \log \Big(\sum_{i \in S} \boldsymbol{p}_i \Big) - \log \Big(\sum_{i \in S} \boldsymbol{q}_i \Big) \right| \leq \frac{1}{\sum_{i \in S} \boldsymbol{p}_i} \left| \sum_{i \in S} (\boldsymbol{p}_i - \boldsymbol{q}_i) \right| \leq \frac{k}{\epsilon} \|\boldsymbol{p} - \boldsymbol{q}\|_1 ,$$

where we used the fact that, since $\boldsymbol{p} \in \widetilde{\Delta}^k$, it holds $\boldsymbol{p}_i \geq \epsilon/k$. Therefore, when we round each coordinate of a vector $\boldsymbol{p}$ to the closest integer multiple of $O((\epsilon^{3/2}\alpha/k)^2)$ we get a vector $\boldsymbol{p}' \in \mathcal{C}$ such that for any set S it holds $|\log(\sum_{i \in S} \boldsymbol{p}_i) - \log(\sum_{i \in S} \boldsymbol{q}_i)| \leq \epsilon^2\alpha^2/6$ which implies that the empirical log-likelihood satisfies $|\mathcal{L}_N(\boldsymbol{p}) - \mathcal{L}_N(\boldsymbol{p}')| \leq \epsilon^2\alpha^2/6$. We will now show that, with high probability, any element $\boldsymbol{p}$ of the cover

$\mathcal{C}$ such that $\|\boldsymbol{p} - \boldsymbol{p}^\star\|_1 \geq \epsilon$, satisfies $\mathcal{L}_N(\boldsymbol{p}^\star) - \mathcal{L}_N(\boldsymbol{p}) \geq \epsilon^2\alpha^2/2$. We will use the following concentration result on likelihood ratios.

Lemma 4.1.6 (Proposition 7.27 of (Mas07))**.** *Let $\mathcal{D}_1, \mathcal{D}_2$ be two distributions (on any domain) with positive density functions f, g respectively. For any $x \in \mathbb{R}$, it holds*

$$\underset{x_1,\ldots,x_N\sim\mathcal{D}_1}{\mathbf{Pr}}\left[\frac{1}{N}\sum_{n=1}^{N}\log\frac{f(x_n)}{g(x_n)} \leq (d_{\mathrm{TV}}(\mathcal{D}_1, \mathcal{D}_2))^2 - 2x/N\right] \leq e^{-x}\,.$$

Using the above lemma with $x = O(\log(|\mathcal{C}|/\delta)) = O(k\log(k/(\epsilon\delta)))$ and

$$N = \Theta(k\log(k/(\epsilon\delta))/(\alpha^2\epsilon^2))\,,$$

we obtain that, with probability at least $1 - \delta/|\mathcal{C}|$, it holds $\mathcal{L}_N(\boldsymbol{p}^\star) - \mathcal{L}_N(\boldsymbol{p}) \geq d_{\mathrm{TV}}(D_\pi, D_\pi^\star)^2 - \alpha^2\epsilon^2/2$. From the union bound, we obtain that the same is true for all vectors $\boldsymbol{p} \in \mathcal{C}$ with probability at least $1-\delta$. We are now ready to finish the proof of the claim. Let $\boldsymbol{p} \in \widetilde{\Delta}^k$ be any probability vector such that $\|\boldsymbol{p} - \boldsymbol{p}^\star\|_1 \geq \epsilon$. Let $\bar{\boldsymbol{p}} \in \widetilde{\Delta}^k$ be the maximizer of the empirical likelihood constrained on $\widetilde{\Delta}^k$, i.e., $\bar{\boldsymbol{p}} = \arg\max_{\boldsymbol{q}\in\widetilde{\Delta}^k}\mathcal{L}_N(\boldsymbol{q})$ and let $\widetilde{\boldsymbol{p}}^\star$ be the closest vector of the cover $\mathcal{C}$ to $\boldsymbol{p}^\star$. We have

$$\mathcal{L}_N(\bar{\boldsymbol{p}}) - \mathcal{L}_N(\boldsymbol{p}) \geq \mathcal{L}_N(\widetilde{\boldsymbol{p}}^\star) - \mathcal{L}_N(\boldsymbol{p}) \geq \mathcal{L}_N(\boldsymbol{p}^\star) - \epsilon^2\alpha^2/6 - \mathcal{L}_N(\boldsymbol{p})\,.$$

The first inequality holds since both $\bar{\boldsymbol{p}}$ and $\widetilde{\boldsymbol{p}}^\star$ lie in $\widetilde{\Delta}^k$. The second inequality holds since we can replace the point of the cover $\widetilde{\boldsymbol{p}}^\star \in \mathcal{C}$, with each closest point in the simplex $\boldsymbol{p}^\star$ without affecting the likelihood value by a lot. Finally, since $\boldsymbol{p}$ lies in $\widetilde{\Delta}^k$, we can replace it with a point $\boldsymbol{p}'$ in the cover with $\|\boldsymbol{p}' - \boldsymbol{p}^\star\|_1 \geq \epsilon$, and get that

$$\mathcal{L}_N(\bar{\boldsymbol{p}}) - \mathcal{L}_N(\boldsymbol{p}) \geq \mathcal{L}_N(\boldsymbol{p}^\star) - \epsilon^2\alpha^2/6 - \mathcal{L}_N(\boldsymbol{p}') - \epsilon^2\alpha^2/6\,,$$

and, since $\mathcal{L}_N(\boldsymbol{p}^\star) - \mathcal{L}_N(\boldsymbol{p}') \geq \epsilon^2\alpha^2/2$, we have that $\mathcal{L}_N(\bar{\boldsymbol{p}}) - \mathcal{L}_N(\boldsymbol{p}) = \Omega(\epsilon^2\alpha^2)$. □

This concludes the proof of Proposition 4.1.5. □

The Proof of Theorem 4.1.3

In this subsection, we prove Theorem 4.1.3. Our goal is to simulate a statistical query oracle which takes as input a query function q with domain $\mathcal{X} \times \mathcal{Z}$ and outputs an estimate of its expectation with respect to finely labeled examples $\mathbb{E}_{(x,z)\sim\mathcal{D}}[q(x,z)]$, using coarsely labeled examples. Recall that since we have sample access only to coarsely labeled examples $(x, S) \sim \mathcal{D}_\pi$, we cannot directly estimate this expectation. The key idea is to perform rejection sampling on each coarse sample (x, S) with acceptance probability $q(x, j)$ for any fine label $j \in \mathcal{Z}$. Because of the rejection sampling process, this marginal distribution is not the marginal of $\mathcal{D}$ on the fine labels $\mathcal{Z}$, but the marginal of $\mathcal{D}$ on the fine labels, conditional on the accepted samples. However, the task of estimating from this marginal distribution can be still reduced to the unsupervised problem (see Proposition 4.1.5) of the previous section. Consider an arbitrary query function $q : \mathcal{X} \times \mathcal{Z} \to [-1, 1]$ and, without loss of generality, let $\mathcal{Z} = [k]$. Recall that $\mathcal{D}$ is the joint probability distribution on the finely labeled examples (x, z). We have that

$$\underset{(x,z)\sim\mathcal{D}}{\mathbb{E}}[q(x,z)] = \sum_{j=1}^{k} \underset{(x,z)\sim\mathcal{D}}{\mathbb{E}}\left[q(x,j)\mathbf{1}\{z=j\}\right] = \sum_{j=1}^{k} \underset{(x,z)\sim\mathcal{D}}{\mathbb{E}}\left[q_j(x)\mathbf{1}\{z=j\}\right]. \tag{4.4}$$

Since we would like to estimate the expectation of the query $q(x, z)$ with tolerance τ, it suffices to estimate the expectation of each query $q_j(x)\mathbf{1}\{z = j\}$ with tolerance τ/k for any $j \in [k]$. Hence, it suffices to estimate expectations of the form $\mathbb{E}_{(x,z)\sim\mathcal{D}}[f(x)\mathbf{1}\{z = j\}]$ for arbitrary functions $f : \mathcal{X} \to [0, 1]$[2] and $j \in [k]$.

Let $\mathcal{D}_x$ denote the marginal distribution of the examples $x \in \mathcal{X}$. The algorithm performs rejection sampling. Each coarsely labeled example $(x, S) \sim \mathcal{D}_\pi$ is accepted with probability $f(x)$, that does not depend on the coarse label S. Hence, the rejection sampling process induces a distribution $\mathcal{D}^f$ over finely labeled examples $(x, z) \in \mathcal{X} \times \mathcal{Z}$ with density

$$\mathcal{D}^f(x,z) = \frac{f(x)}{\mathbb{E}_{x\sim\mathcal{D}_x}[f(x)]}\mathcal{D}(x,z).$$

We remark that, we do not have sample access to $\mathcal{D}^f$ because we do not have sample access to the distribution $\mathcal{D}$ of the fine examples; we introduced the above notation for the purposes of the proof. Similarly, to $\mathcal{D}_x$, we define $\mathcal{D}_x^f$ to be the marginal distribution of x conditional on its acceptance, i.e.,

$$\mathcal{D}_x^f(x) = \frac{f(x)}{\mathbb{E}_{x\sim\mathcal{D}_x}[f(x)]}\mathcal{D}_x(x). \tag{4.5}$$

Let $\mathcal{D}_z$ denote the marginal distribution of the fine labels $[k]$ and let $\mathcal{D}_z(\cdot|x)$ be the marginal distribution conditional on the example x. We have that

$$\underset{(x,z)\sim\mathcal{D}}{\mathbb{E}}\left[f(x)\mathbf{1}\{z=j\}\right] = \int_{\mathcal{X}} f(x)\mathcal{D}(x,j)dx = \int_{\mathcal{X}} f(x)\mathcal{D}_x(x)\mathcal{D}_z(j|x)dx.$$

The above expectation can be equivalently written, by multiplying and dividing by $\mathcal{D}_x^f$,

$$\underset{(x,z)\sim\mathcal{D}}{\mathbb{E}}\left[f(x)\mathbf{1}\{z=j\}\right] = \int_{\mathcal{X}} \left(\frac{f(x)\mathcal{D}_x(x)}{\mathcal{D}_x^f(x)}\right)\left(\mathcal{D}_x^f(x)\mathcal{D}_z(j|x)\right)dx.$$

The first term in the integral is equal to $\mathbb{E}_{x\sim\mathcal{D}_x}[f(x)]$, by substituting Equation (4.5) and, hence, is constant. The second term corresponds to the probability of observing the fine label j, given an example x, that has been accepted from the rejection

[2] Any function $f : \mathcal{X} \to [-1, 1]$ can be decomposed into $f = f^+ - f^-$ with $f^+, f^- \geq 0$ and, by linearity of expectation, it suffices to work with functions f with image in $[0, 1]$.

sampling process. Similarly, to the marginal $\mathcal{D}_z$, we define $\mathcal{D}_z^f$ to be the marginal distribution of the fine labels z conditional on acceptance. Hence, we can write

$$\mathop{\mathbb{E}}_{(x,z)\sim\mathcal{D}}\Big[f(x)\mathbf{1}\{z=j\}\Big] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}[f(x)] \cdot \mathop{\mathbf{Pr}}_{z\sim\mathcal{D}_z^f}[z=j]\,. \tag{4.6}$$

The decomposition of the expectation of Equation (4.6) is a key step: we now only need to learn the marginal distribution of fine labels conditional on acceptance $\mathcal{D}_z^f$.

Recall that our goal is to estimate the left hand side expectation of Equation (4.6) with tolerance τ/k. We claim that it suffices to estimate each term of the right hand side product of Equation (4.6) with tolerance $\tau/(2k)$. This is implied from the following: consider an estimate $\widetilde{\mu}$ of the value $\mathbb{E}_{x\sim\mathcal{D}_x}[f(x)]$ and an estimate $\widetilde{p}$ of the value $\mathbf{Pr}_{z\sim\mathcal{D}_z^f}[z=j]$. Then, using Equation (4.6), we have that

$$\left|\widetilde{\mu}\cdot\widetilde{p} - \mathop{\mathbb{E}}_{(x,z)\sim\mathcal{D}}[f(x)\mathbf{1}\{z=j\}]\right| = \left|\widetilde{\mu}\cdot\widetilde{p} - \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}[f(x)]\cdot\mathop{\mathbf{Pr}}_{z\sim\mathcal{D}_z^f}[z=j]\right|,$$

and, hence, by adding and subtracting the term $\widetilde{\mu}\ \mathbf{Pr}_{z\sim\mathcal{D}_z^f}[z=j]$, using the triangle inequality and, since both $\mathbb{E}_{x\sim\mathcal{D}_x}[f(x)]$ and $\mathbf{Pr}_{z\sim\mathcal{D}_z^f}[z=j]$ are at most 1, we get that

$$\left|\widetilde{\mu}\cdot\widetilde{p} - \mathop{\mathbb{E}}_{(x,z)\sim\mathcal{D}}[f(x)\mathbf{1}\{z=j\}]\right| \le \left|\widetilde{\mu} - \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}[f(x)]\right| + \left|\widetilde{p} - \mathop{\mathbf{Pr}}_{z\sim\mathcal{D}_z^f}[z=j]\right|.$$

We will show that $O(k^4/(\tau^3\alpha^2)\log(1/\delta))$ samples are sufficient to bound each term of the right hand side by $\tau/(2k)$, with high probability. In order to estimate the expectation $\mathbb{E}_{(x,z)\sim\mathcal{D}}[q(x,z)]$, the algorithm applies (in parallel) the above process k times with $f=q_j$ for any $j\in[k]$ (using Equation (4.4)) using a single training set of size $N=O(k^4/(\tau^3\alpha^2)\log(1/\delta))$ drawn from the distribution $\mathcal{D}_\pi$ of coarsely labeled examples. Moreover, the running time is polynomial in the number of samples N. To conclude the proof, it suffices to show the following claims.

Claim 2. *There exists an algorithm that, uses $N=\widetilde{O}(k^4/(\tau^3\alpha^2)\log(1/\delta))$ samples from $\mathcal{D}_\pi$ and computes an estimate $\widetilde{p}$, that satisfies $\left|\widetilde{p}-\mathbf{Pr}_{z\sim\mathcal{D}_z^f}[z=j]\right|\le\tau/(2k)$, with probability at least $1-\delta$.*

Proof. Recall that the distribution $\mathcal{D}_z^f$ is the marginal distribution of the fine labels $z\in\mathcal{Z}=[k]$, conditional that the example $x\sim\mathcal{D}_x^f$, i.e., that the example $x\in\mathcal{X}$ has been accepted by the rejection sampling process. Hence, the distribution $\mathcal{D}_z^f$ is supported on $\mathcal{Z}$. We can then directly apply Proposition 4.1.5, using as training set the set of *accepted* coarsely labeled samples (x,S) and can compute an estimate $\widetilde{\mathcal{D}}$, that is ϵ-close in total variation distance to $\mathcal{D}_z^f$. By setting $\epsilon=\tau/(2k)$, the algorithm uses $\widetilde{O}(k^3/(\tau^2\alpha^2)\log(1/\delta))$ samples from the set of accepted samples and outputs the estimate $\widetilde{p}=\widetilde{\mathcal{D}}(j)$. For the example $x\in\mathcal{X}$, the acceptance probability $f(x)$ can be considered $\Omega(\tau/k)$. Otherwise, we can set the desired expectation equal to 0. Hence, the algorithm needs to draw in total $\widetilde{O}(k^4/(\tau^3\alpha^2)\log(1/\delta))$ samples from $\mathcal{D}_\pi$ in order to compute an estimate $\widetilde{p}$ that satisfies

$$\left|\widetilde{p} - \mathop{\mathbf{Pr}}_{z\sim\mathcal{D}_z^f}[z = j]\right| \leq \tau/(2k)\,,$$

with probability at least $1-\delta$. □

Claim 3. *There exists an algorithm that, uses $N = O((k^2/\tau^2)\log(1/\delta))$ samples from $\mathcal{D}_\pi$ and computes an estimate $\widetilde{\mu}$, that satisfies $\left|\widetilde{\mu} - \mathbb{E}_{x\sim\mathcal{D}_x}[f(x)]\right| \leq \tau/(2k)\,$, with probability at least $1-\delta$.*

Proof. The algorithms draws N coarsely labeled examples from $\mathcal{D}_\pi$ and computes the estimate $\widetilde{\mu} = \frac{1}{N}\sum_{i=1}^N f(x_i)$. From the Hoeffding bound, since the estimate is a sum of independent bounded random variables, we get

$$\mathbf{Pr}\left[\left|\widetilde{\mu} - \mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}[f(x)]\right| \geq \tau/(2k)\right] \leq 2\exp(-N\tau^2/(2k^2))\,.$$

Using $N = O((k^2/\tau^2)\log(1/\delta))$ samples, the algorithm estimates the desired expectation with error $\tau/(2k)$, with probability at least $1-\delta$. Note that, if $\widetilde{\mu} < \tau/(2k)$, the algorithm can output 0, since the estimated value will lie in the desired tolerance interval. □

4.1.3 Training Models from Coarse Data

Consider a parameterized family of functions $\boldsymbol{x} \to f(\boldsymbol{x};\boldsymbol{w})$, where the parameters $\boldsymbol{w}$ lie in some parameter space $\mathcal{W} \subseteq \mathbb{R}^p$. For instance, the family may correspond to a feed-forward neural network with L layers. Given a finely labeled training sample $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N) \in \mathcal{X}\times\mathcal{Y}$, the parameters $\boldsymbol{w}$ are chosen using a gradient method in order to minimize the empirical risk,

$$\mathcal{L}_N(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^N \ell(f(\boldsymbol{x}_i;\boldsymbol{w}), y_i)\,,$$

for some loss function $\ell : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$ and the goal of this optimization task is to minimize the population risk function $\mathcal{L}(\boldsymbol{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}(\boldsymbol{w}^\star)}[\ell(f(\boldsymbol{x};\boldsymbol{w}), y)]$ (where the distribution $\mathcal{D}(\boldsymbol{w}^\star)$ is unknown). For simplicity, let us focus on differentiable loss functions. Performing the SGD algorithm, we can circumvent the lack of knowledge of the population risk function $\mathcal{L}$. Specifically, instead of computing the gradient of $\mathcal{L}(\boldsymbol{w})$, the algorithm steps towards a random direction $\boldsymbol{v}$ with the constraint that the expected value of $\boldsymbol{v}$ is equal to the negative of the true gradient, i.e., it is an unbiased estimate of $-\nabla\mathcal{L}(\boldsymbol{w})$. Such a random vector $\boldsymbol{v}$ can be computed without knowing $\mathcal{D}(\boldsymbol{w}^\star)$ using the interchangeability between the expectation and the gradient operators. Assume that the algorithm is at iteration

$t \geq 1$. Let $(\boldsymbol{x}, y) \sim \mathcal{D}(\boldsymbol{w}^\star)$ be a fresh sample and define $\boldsymbol{v}_t$ be the gradient of the loss function with respect to $\boldsymbol{w}$, at the point $\boldsymbol{w}_t$, i.e.,

$$\mathbb{E}[\boldsymbol{v}_t|\boldsymbol{w}_t] = \underset{(\boldsymbol{x},y)\sim\mathcal{D}(\boldsymbol{w}^\star)}{\mathbb{E}}[\nabla\ell(f(\boldsymbol{x};\boldsymbol{w}_t), y)] = \nabla \underset{(\boldsymbol{x},y)\sim\mathcal{D}(\boldsymbol{w}^\star)}{\mathbb{E}}[\ell(f(\boldsymbol{x};\boldsymbol{w}_t), y)] = \nabla\mathcal{L}(\boldsymbol{w}_t)\,.$$

Hence, an algorithm that has query access to a SQ oracle can implement a noisy version of the above iterative process (with inexact gradients, see e.g., (d'A08; DGN14; FGV17)) using the query functions $q_i(\boldsymbol{x}, y) = (\nabla\ell(f(\boldsymbol{x};\boldsymbol{w}_t), y))_i$ for any $i \in [p]$. Note that the algorithm knows the loss function ℓ, the parameterized functions' family $\{f(\cdot\,;\boldsymbol{w}) : \boldsymbol{w} \in \mathcal{W}\}$ and the current guess $\boldsymbol{w}_t$. Specifically, the algorithm performs p queries (one for each coordinate of the parameter vector) and the oracle returns to the algorithm a noisy gradient vector $\boldsymbol{r}_t$ that satisfies $\|\boldsymbol{r}_t - \nabla\mathcal{L}(\boldsymbol{w}_t)\|_\infty \leq \tau$.

In our setting, we do not have access to the SQ oracle with finely labeled examples. Our main result of this section (Theorem 4.1.3) is a mechanism that enables us to obtain access to such an oracle using a few coarsely labeled examples (with high probability). Hence, we can still perform the noisy gradient descent of the previous paragraph with an additional overhead on the sample complexity, due to the reduction.

4.1.4 Multiclass Logistic Regression with Coarsely Labeled Data

A first application for the above generic reduction from coarse data to statistical queries is the case of coarse multiclass logistic regression. In the standard (finely labeled) multiclass logistic regression problem, there are k fine labels (that correspond to classes), each one associated with a weight vector $\boldsymbol{w}_z \in \mathbb{R}^n$ with $z \in [k]$. We can consider the weight matrix $\boldsymbol{W} \in \mathbb{R}^{k\times n}$. Given an example $\boldsymbol{x} \in \mathbb{R}^n$, the vector $\boldsymbol{x}$ is filtered via the softmax function $\sigma(\boldsymbol{W}, \boldsymbol{x})$, which is a probability distribution over Δ^k with $\sigma(\boldsymbol{W}, \boldsymbol{x}; z) = \exp(\boldsymbol{w}_z^T\boldsymbol{x})/\sum_{y\in[k]}\exp(\boldsymbol{w}_y^T\boldsymbol{x})$, $z \in [k]$ and the output is the finely labeled example $(\boldsymbol{x}, z) \in \mathbb{R}^n \times [k]$. The goal is to estimate the weight matrix $\boldsymbol{W}$, given finely labeled examples. Let us denote by $\mathcal{D}(\boldsymbol{W})$ the joint distribution over the finely labeled examples for simplicity. When we have access to finely labeled examples $(\boldsymbol{x}, z) \sim \mathcal{D}(\boldsymbol{W}^\star)$, the population log-likelihood objective $\mathcal{L}$ of the multiclass logistic regression problem

$$\mathcal{L}(\boldsymbol{W}) = \underset{(\boldsymbol{x},z)\sim\mathcal{D}(\boldsymbol{W}^\star)}{\mathbb{E}}\left[\boldsymbol{w}_z^T\boldsymbol{x} - \log\left(\sum_{j\in\mathcal{Z}}\exp(\boldsymbol{w}_j^T\boldsymbol{x})\right)\right],$$

is concave (see (FHT01)) with respect to the weight matrix $\boldsymbol{W} \in \mathbb{R}^{k\times n}$ and is solved using gradient methods. On the other hand, if we have sample access only to coarsely labeled examples $(\boldsymbol{x}, S) \sim \mathcal{D}_\pi(\boldsymbol{W}^\star)$, the population log-likelihood objective $\mathcal{L}_\pi$ of the coarse multiclass logistic regression problem

$$\mathcal{L}_\pi(\boldsymbol{W}) = \underset{(\boldsymbol{x},S)\sim\mathcal{D}_\pi(\boldsymbol{W}^\star)}{\mathbb{E}}\left[\log\left(\sum_{z\in S}\exp(\boldsymbol{w}_z^T\boldsymbol{x})\right) - \log\left(\sum_{j\in\mathcal{Z}}\exp(\boldsymbol{w}_j^T\boldsymbol{x})\right)\right],$$

which is no more concave. However, as an application of our main result (Theorem 4.1.3), we can still solve it. In fact, since we can implement statistical queries using the sample access to the coarse data generative process $\mathcal{D}_\pi(\boldsymbol{W}^\star)$, we can compute the gradients of the log-likelihood objective that corresponds to the *finely labeled examples*. Hence, the total sample complexity of optimizing this non-convex objective is equal to the sample complexity of solving the convex problem with an additional overhead at each iteration of computing the gradients, that is given by Theorem 4.1.3.

4.2 Gaussian Mean Estimation from Coarse Data

In this section, we consider the fundamental problem of efficiently learning coarse Gaussian distributions in high dimensions. Lately, significant progress has been made from a computational point of view in such censored/truncated settings in the distribution specific setting, e.g., when the underlying distribution is Gaussian (DGTZ18; KTZ19), mixtures of Gaussians (NP19), linear regression (DGTZ19b; IZD20; DRZ20). In this distribution specific setting, we consider the most fundamental problem of learning the mean of a Gaussian distribution given coarse data. Let us recall the generative process to the reader.

Definition 4.2.1 (Coarse Gaussian Data)**.** *Consider the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^\star)$, with mean $\boldsymbol{\mu}^\star \in \mathbb{R}^d$ and identity covariance matrix. We generate a sample as follows: (i) Draw $\boldsymbol{z}$ from $\mathcal{N}(\boldsymbol{\mu}^\star)$, (ii) Draw a partition $\mathcal{S}$ (of $\mathbb{R}^d$) from π. Finally, we observe the set $S \in \mathcal{S}$ that contains $\boldsymbol{z}$. We denote the distribution of S as $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$.*

We remark that we only require membership oracle access to the subsets of the partition $\mathcal{S}$. A set $S \subseteq \mathbb{R}^d$ corresponds to a membership oracle $\mathcal{O}_S : \mathbb{R}^d \to \{0, 1\}$ that given $\boldsymbol{x} \in \mathbb{R}^d$ outputs whether the point lies inside the set S or not.

The main results of this section can be summarized in the following two statements.

Theorem 4.2.2 (Hardness of Matching the Observed Distribution with General Partitions)**.** *Let π be a general partition distribution. Unless* $\mathrm{P} = \mathrm{NP}$*, no algorithm with sample access to $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, can compute, in* $\mathrm{poly}(d)$ *time, a $\widetilde{\boldsymbol{\mu}} \in \mathbb{R}^d$ such that $d_{\mathrm{TV}}(\mathcal{N}_\pi(\widetilde{\boldsymbol{\mu}}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star)) < 1/d^c$ for some absolute constant $c > 1$.*

We prove our hardness result using a reduction from the well known Max-Cut problem, which is known to be NP-hard, even to approximate (Hås01). In our reduction, we use partitions that consist of simple sets: fat hyperplanes, ellipsoids and their complements: the computational hardness of this problem is rather inherent and not due to overly complicated sets.

On the positive side, we identify a geometric property that enables us to design a computationally efficient algorithm for this problem: Namely we require all the sets of the partitions to be *convex*.

Theorem 4.2.3 (Gaussian Mean Estimation with Convex Partitions)**.** *Let $\epsilon, \delta \in (0,1)$. Consider the generative process of coarse d-dimensional Gaussian data $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, as in Definition 4.2.1. Assume that the partition distribution π is α-information preserving and is supported on convex partitions of $\mathbb{R}^d$. The following hold.*

1. *The empirical log-likelihood objective*

$$\mathcal{L}_N(\boldsymbol{\mu}) = \frac{1}{N}\sum_{i=1}^{N} \log \mathcal{N}(\boldsymbol{\mu}; S_i)$$

   is concave with respect to $\boldsymbol{\mu}$ where the sets S_i for $i \in [N]$ are i.i.d. samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$.

2. *There exists an algorithm, that draws $N = \widetilde{O}(d/(\epsilon^2\alpha^2)\log(1/\delta))$ samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$ and computes an estimate $\widetilde{\boldsymbol{\mu}}$ that satisfies $d_{\mathrm{TV}}(\mathcal{N}(\widetilde{\boldsymbol{\mu}}), \mathcal{N}(\boldsymbol{\mu}^\star)) \leq \epsilon$, with probability at least $1-\delta$.*

4.2.1 The Proofs of Theorem 4.2.2 & 4.2.3

In this section, we present the proofs of our results regarding the fundamental problem of learning a Gaussian distribution given coarse data. In Section 4.2.2, we show that, under general partitions, this problem is NP-hard. In Section 4.2.3, we show that we can efficiently estimate the Gaussian mean under convex partitions of the space.

4.2.2 Computational Hardness under General Partitions

In this section, we consider general partitions of the d-dimensional Euclidean space, that may contain non-convex subsets. For instance, a compact convex body and its complement define a non-convex partition of $\mathbb{R}^d$. In order to get this computational hardness result, we reduce from Max-Cut and make use of its hardness of approximation (see (Hås01)). Recall that Max-Cut can be viewed as a maximization problem, where the objective function corresponds to a particular quadratic function (associated with the Laplacian graph of the given graph instance) and the constraints restrict the solution to lie in the Boolean hypercube (the constraints can be seen geometrically as the intersection of bands, see Figure 4.1).

We first define Max-Cut and a variant of Max-Cut where the optimal cut score is given as part of the input. Let $G = (V, E)$ be a graph[3] with d vertices. A *cut* is a partition of V into two subsets S and $S' = V \setminus S$ and the value of the cut (S, S') is $c(S, S') = \sum_{u,v \in E} \mathbf{1}\{u \in S, v \in S'\}$. The goal of the problem is find the maximum value cut in G, i.e., to partition the vertices into two sets so that the

[3] We are going to work with graphs with unit weights.

number of edges crossing the cut is maximized. We can define Max-Cut as the following maximization problem for the graph $G = (V, E)$ with $|V| = d$:

$$\max \sum_{(i,j)\in E} (\boldsymbol{x}_i - \boldsymbol{x}_j)^2 \,, \quad \text{subj. to} \;\; \boldsymbol{x}_i \in \{-1, +1\} \; \forall i \in [d] \,.$$

The objective function is the quadratic form $\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x}$, where $\boldsymbol{L}_G$ is the Laplacian matrix of the graph G. We may also assume that the value of the optimal cut is known and is equal to opt.[4] Before proceeding with the overview of the proof, we state a key result of (Hås01) about the inapproximability of Max-Cut .

Lemma 4.2.4 (Inapproximability of Maximum Cut Problem (Hås01))**.** *It is* NP*-hard to approximate* Max-Cut *to any factor higher than* 16/17*.*

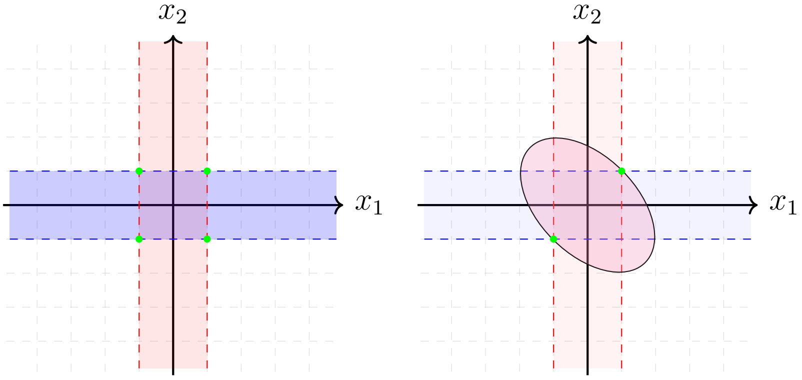


Figure 4.1: The geometry of the Max-Cut instance. The left figure corresponds to the fat hyperplanes, i.e., the constraints of Max-Cut and the right figure (the ellipsoid) corresponds to the objective function of Max-Cut . The green points lie in the Boolean hypercube.

Sketch of the Proof of Theorem 4.2.2

The first step of the proof is to construct the distribution over partitions of $\mathbb{R}^d$. The Max-Cut problem can be viewed as a collection of $d+1$ non-convex partitions of the d-dimensional Euclidean space. Consider an instance of Max-Cut with $|V| = d$ and optimal cut value opt. Consider the collection of $d+1$ partitions $\mathcal{B} = \{\mathcal{S}_1, \ldots, \mathcal{S}_d, \mathcal{T}\}$. We define the partitions as follows: for any $i = 1, \ldots, d$, we let $S_i = \{\boldsymbol{x} : -1 \leq \boldsymbol{x}_i \leq 1\}$ be the sets that correspond to fat hyperplanes of Figure 4.1(a) and the partitions $\mathcal{S}_i = \{S_i, S_i^c\}$, i.e., pairs of fat hyperplanes

[4] Observe that this problem is still hard, since the maximum value of a cut is bounded by d^2 and, hence, if this problem could be solved efficiently, one would be able to solve Max-Cut by trying all possible values of opt.

and their complements (see Figure 4.2(a,b)). These d partitions will simulate the Max-Cut constraints, i.e., that the solution vector lies in the hypercube $\{-1,1\}^d$. It remains to construct $\mathcal{T}$, which intuitively corresponds to the quadratic objective of Max-Cut .

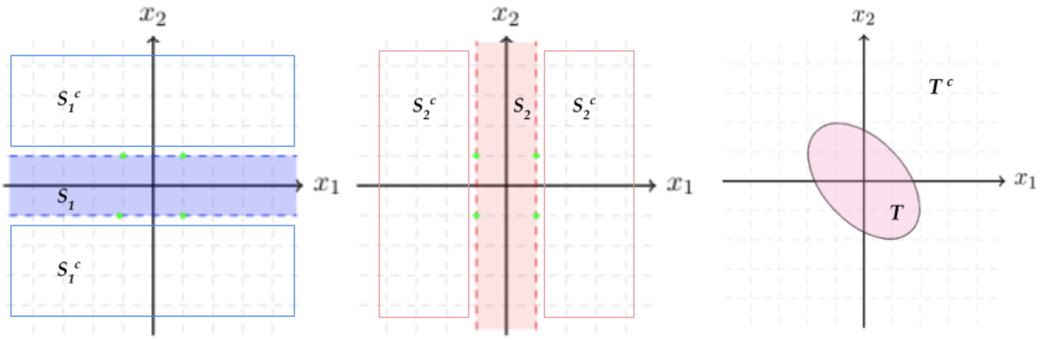


Figure 4.2: The mixture of partitions that corresponds to the Max-Cut problem. In figures (a) and (b), we partition the Euclidean space using fat hyperplanes (the blue set S_1 and the red set S_2 respectively) and their complements $S_1^c = \mathbb{R}^d \backslash S_1$ and $S_2^c = \mathbb{R}^d \backslash S_2$. The third figure (c) partitions $\mathbb{R}^d$ using the ellipsoid $T = \{\boldsymbol{x} : \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \le q\}$ and its complement $T^c = \mathbb{R}^d \setminus T$ (for some $d \times d$ covariance matrix $\boldsymbol{\Sigma}$ and positive real q).

Fix the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{L}_G^{-1}\text{opt}$ [5] , i.e., $\boldsymbol{\Sigma}$ is the inverse of the Laplacian normalized by opt. We let $T = \{\boldsymbol{x} : \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \le q\}$ for some positive value q to be defined later (see Figure 4.1(b) and Figure 4.2(c)). Then, we let $\mathcal{T} = \{T, T^c\}$. We construct a mixture π of these partitions by picking each one uniformly at random, i.e., with probability $1/(d+1)$.

Let us assume that there exists an algorithm that, given access to samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$, with *known covariance* $\boldsymbol{\Sigma}$, computes, in time poly(d), a mean vector $\boldsymbol{\mu}$ so that the output distributions are matched, i.e., $d_{\mathrm{TV}}(\mathcal{N}_\pi(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}))$ is upper bounded by $1/d^c$ for some absolute constant $c > 1$. Equivalently this means that the mass that $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ assigns to each set S_i and T is within poly$(1/d)$ of the corresponding mass that $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$ assigns to the same set. There are two main challenges in order to prove the reduction:

1. How can we generate coarse samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$ since $\boldsymbol{\mu}^\star$ is the solution of the Max-Cut problem and therefore is unknown?

2. Given opt, is it possible to pick the threshold q of the ellipsoid $T = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \le q\}$ so that any vector $\boldsymbol{\mu}$ (rounded to belong in $\{-1,1\}^d$),

[5] In fact, $\boldsymbol{L}_G$ has zero eigenvalue with eigenvector $(1,\ldots,1)$: we have to project the Laplacian to the subspace orthogonal to $(1,\ldots,1)$ to avoid this. We ignore this technicality here for simplicity.

that achieves $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; T) \approx \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; T)$ and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; S_i) \approx \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$, also achieves an approximation ratio better than $16/17$ for the Max-Cut objective ?

The key observation to answer the first question is that, by the rotation invariance of the Gaussian distribution, the probability

$$\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; T) = \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})} \left[\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \le q\right]$$

is a constant p that only depends on the value opt of the Max-Cut problem. Therefore, having this value p, we can flip a coin with this probability and give the coarse sample T if we get heads and T^c otherwise. Similarly, the value of $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$ is an absolute constant that does not depend on $\boldsymbol{\mu}^\star \in \{-1, 1\}^d$ and therefore we can again simulate coarse samples by flipping a coin with probability equal to $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$.

To resolve the second question, we first show that any vector $\boldsymbol{\mu}$ that approximately matches the probabilities of the d fat halfspaces, lies very close to a corner of the hypercube, see Lemma 4.2.7. Therefore, by rounding this guess $\boldsymbol{\mu}$, we obtain exactly a corner of the hypercube without affecting the probability assigned to the ellipsoid constraint by a lot. We then show that any vector of the hypercube that almost matches the probability of the ellipsoid achieves large cut value. In particular, we prove that there exists a value for the threshold q of the ellipsoid $\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \le q$ that makes the probability $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; T)$ *very sensitive to changes of* $\boldsymbol{\mu}$. Therefore, the only way for the algorithm to match the observed probability is to find a $\boldsymbol{\mu}$ that achieves large cut value. We show the following lemma.

Lemma 4.2.5 (Sensitivity of Gaussian Probability of Ellipsoids)**.** *Let $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be d-dimensional Gaussian distributions. Let $\boldsymbol{v}^\star = \boldsymbol{\Sigma}^{-1/2} \boldsymbol{\mu}^\star$, $\boldsymbol{v} = \boldsymbol{\Sigma}^{-1/2} \boldsymbol{\mu}$ and assume that $\|\boldsymbol{v}\|_2 \le \|\boldsymbol{v}^\star\|_2 = 1$. Denote $q = d + \|\boldsymbol{v}^\star\|_2^2 + \sqrt{2d + 4\|\boldsymbol{v}^\star\|_2^2}$. Then, assuming d is larger than some sufficiently large absolute constant, it holds that*

$$\left| \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})} \left[\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \le q\right] - \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \left[\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \le q\right] \right| \ge \frac{\|\boldsymbol{v}^\star\|_2^2 - \|\boldsymbol{v}\|_2^2}{6\sqrt{2d+4}} - o(1/\sqrt{d})\,.$$

Notice that with $\boldsymbol{\Sigma} = \boldsymbol{L}_G^{-1}\mathrm{opt}$, in the above lemma, we have $\|\boldsymbol{v}^\star\|_2^2 = 1$, since $\boldsymbol{\mu}^\star$ achieves cut value opt. By assumption, we know that the learning algorithm can find a guess $\boldsymbol{\mu}$ that makes the left hand side of the inequality of Lemma 4.2.5 smaller than $\mathrm{poly}(1/d)$. Thus, we obtain that, for d large enough, it must be that $\|\boldsymbol{v}\|_2^2 = \boldsymbol{\mu}^T \boldsymbol{L}_G \boldsymbol{\mu}/\mathrm{opt} \ge 16/17$. Therefore, $\boldsymbol{\mu}$ achieves value greater than $(16/17)\mathrm{opt}$.

Remark 2. *The transformation π used in the above hardness result is not information preserving. In Theorem 4.2.2, we prove that it is computationally hard to find a vector $\boldsymbol{\mu} \in \mathbb{R}^d$ that matches in total variation the* observed distribution over coarse labels*. In contrast, as we will see in the upcoming Section 4.2.3, when the sets of the partitions are convex, we show that there is an efficient algorithm that can* *solve the same problem and compute some* $\boldsymbol{\mu} \in \mathbb{R}^d$ *such that* $\mathrm{TV}(\mathcal{N}_\pi(\boldsymbol{\mu}^\star), \mathcal{N}_\pi(\boldsymbol{\mu}))$ *is small regardless of whether the transformation* π *is information preserving. When the transformation is information preserving, we can further show that the vector* $\boldsymbol{\mu}$ *that we compute will be close to* $\boldsymbol{\mu}^\star$*.*

Sensitivity of Gaussian Probabilities

We now prove Lemma 4.2.5, namely that the probability of an ellipsoid with respect to the Gaussian distribution is sensitive to small changes of its mean.

Proof of Lemma 4.2.5. We first observe that

$$\begin{aligned}\underset{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}{\mathbf{Pr}}\left[\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \le q\right] &= \underset{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}{\mathbf{Pr}}\left[\boldsymbol{x}^T\boldsymbol{x} + 2\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x} \le q - \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right] \\ &= \underset{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}{\mathbf{Pr}}\left[\boldsymbol{x}^T\boldsymbol{x} + 2\boldsymbol{v}^T\boldsymbol{x} \le q - \|\boldsymbol{v}\|_2^2\right],\end{aligned}$$

where $\boldsymbol{v} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}$. Similarly, we have

$$\underset{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}{\mathbf{Pr}}\left[\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \le q\right] = \underset{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}{\mathbf{Pr}}\left[\boldsymbol{x}^T\boldsymbol{x} + 2(\boldsymbol{v}^\star)^T\boldsymbol{x} \le q - \|\boldsymbol{v}^\star\|_2^2\right],$$

where $\boldsymbol{v}^\star = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}^\star$. From the rotation invariance of the Gaussian distribution, we may assume, without loss of generality, that $\boldsymbol{v} = \|\boldsymbol{v}\|\boldsymbol{e}_1$ and $\boldsymbol{v}^\star = \|\boldsymbol{v}^\star\|\boldsymbol{e}_1$. Notice that $(\|\boldsymbol{v}\|_2 + \boldsymbol{x}_1)^2 + \sum_{i=2}^d \boldsymbol{x}_i^2$ is a sum of independent random variables. To estimate these probabilities we are going to use the central limit theorem.

Lemma 4.2.6 (CLT, Theorem 1, Chapter XVI in (Fel57))**.** *Let* $X_1, \ldots, X_n$ *be independent random variables with* $\mathbb{E}[|X_i|^3] < +\infty$ *for all* i*. Let* $m_1 = \mathbb{E}[\sum_{i=1}^n X_i]$ *and* $m_j = \sum_{i=1}^n \mathbb{E}[(X_i - \mathbb{E}[X_i])^j]$*. Then,*

$$\mathbf{Pr}\left[\frac{(\sum_{i=1}^n X_i) - m_1}{\sqrt{m_2}} \le x\right] - \Phi(x) = m_3\frac{(1-x^2)\phi(x)}{6m_2^{3/2}} + o\left(n/m_2^{3/2}\right),$$

where $\Phi(\cdot)$*, resp.,* $\phi(\cdot)$ *is the CDF resp., PDF of the standard normal distribution and the convergence is uniform for all* $x \in \mathbb{R}$*.*

Using the above central limit theorem we obtain

$$\underset{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}{\mathbf{Pr}}\left[(\|\boldsymbol{v}^\star\|_2 + \boldsymbol{x}_1)^2 + \sum_{i=2}^d \boldsymbol{x}_i^2 \le q\right] = \Phi(\bar{q}_1) + O\left(\frac{1}{\sqrt{d}}\right)(1-\bar{q}_1{}^2)\phi(\bar{q}_1) + o\left(1/\sqrt{d}\right),$$

where $\bar{q}_1 = \frac{q-(d+\|\boldsymbol{v}^\star\|_2^2)}{\sqrt{2d+4\|\boldsymbol{v}^\star\|_2^2}}$. Since $q = d + \|\boldsymbol{v}^\star\|^2 + \sqrt{2d + 4\|\boldsymbol{v}^\star\|_2^2}$ we obtain $\bar{q}_1 = 1$ and therefore

$$\underset{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}{\mathbf{Pr}}\left[\boldsymbol{x}^T\boldsymbol{x} + 2(\boldsymbol{v}^\star)^T\boldsymbol{x} \le q - \|\boldsymbol{v}^\star\|_2^2\right] = \Phi(1) + o\left(1/\sqrt{d}\right).$$

Similarly, from the central limit theorem, we obtain

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[(\|\boldsymbol{v}\|_2+\boldsymbol{x}_1)^2+\sum_{i=2}^{d}\boldsymbol{x}_i^2\le q\right]=\Phi(\bar{q}_2)+O\left(\frac{1}{\sqrt{d}}\right)(1-\bar{q}_2{}^2)\phi(\bar{q}_2)+o\left(1/\sqrt{d}\right),$$

where $\bar{q}_2=\frac{q-(d+\|\boldsymbol{v}\|_2^2)}{\sqrt{2d+4\|\boldsymbol{v}\|_2^2}}=1+O(1/\sqrt{d})$. Therefore, we have

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\boldsymbol{x}^T\boldsymbol{x}+2\boldsymbol{v}^T\boldsymbol{x}\le q-\|\boldsymbol{v}\|_2^2\right]=\Phi(\bar{q}_2)+o\left(1/\sqrt{d}\right).$$

Moreover, we have that $\bar{q}_2\ge 1+(\|\boldsymbol{v}^\star\|_2^2-\|\boldsymbol{v}\|_2^2)/(\sqrt{2d+4\|\boldsymbol{v}\|_2^2})$. Using the fact that d is sufficiently large and standard approximation results on the Gaussian CDF, we obtain

$$\Phi\left(1+\frac{\|\boldsymbol{v}^\star\|_2^2-\|\boldsymbol{v}\|_2^2}{\sqrt{2d+4\|\boldsymbol{v}\|_2^2}}\right)-\Phi(1)\ge(\|\boldsymbol{v}^\star\|_2^2-\|\boldsymbol{v}\|_2^2)/\left(6\sqrt{2d+4\|\boldsymbol{v}\|_2^2}\right),$$

and, since $\|\boldsymbol{v}\|_2\le 1$, we conclude that the left-hand side satisfies

$$\Phi\left(1+\frac{\|\boldsymbol{v}^\star\|_2^2-\|\boldsymbol{v}\|_2^2}{\sqrt{2d+4\|\boldsymbol{v}\|_2^2}}\right)-\Phi(1)\ge(\|\boldsymbol{v}^\star\|_2^2-\|\boldsymbol{v}\|_2^2)/\left(6\sqrt{2d+4}\right).$$

The result follows. □

We will also require the following sensitivity lemma about the Gaussian probability of bands, i.e., sets of the form $\{\boldsymbol{x}:|\boldsymbol{x}_i|\le 1\}$. We show that the probabilities of such regions are also sensitive under perturbations of the mean of the Gaussian. This means that any vector $\boldsymbol{\mu}$ that has $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[-1\le\boldsymbol{x}_i\le 1\right]$ close to $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}\left[-1\le\boldsymbol{x}_i\le 1\right]$ must be very close to a corner of the hypercube.

Lemma 4.2.7 (Sensitivity of Gaussian Probability of Bands)**.** *Let* $\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma}),\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ *be two d-dimensional Gaussian distributions with* $\boldsymbol{e}_i^T\boldsymbol{\Sigma}\boldsymbol{e}_i\le Q$*, and* $|\boldsymbol{\mu}_i^\star|=1$ *for all* $i\in[d]$*. Then, for any* $i\in[d]$*, it holds that*

$$\left|\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}\left[-1\le\boldsymbol{x}_i\le 1\right]-\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[-1\le\boldsymbol{x}_i\le 1\right]\right|\ge c\cdot\frac{\min(1,(1-|\boldsymbol{\mu}_i|)^2)}{Q^4},$$

for some absolute constant $c\in(0,1]$*.*

Proof. Let us fix $i\in[d]$, define $\mu^\star$ (resp. μ) for $\boldsymbol{\mu}_i^\star$ (resp. $\boldsymbol{\mu}_i$), and $\sigma^2=\boldsymbol{\Sigma}_{ii}$. Without loss of generality since both Gaussians have the same variance σ by symmetry we may assume that $\mu^\star=1$ and $\mu\in[0,+\infty)$. We first deal with the case $\mu>1$. We have

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}&\left[-1\le\boldsymbol{x}_i\le 1\right]-\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[-1\le\boldsymbol{x}_i\le 1\right]\\&=\mathop{\mathbb{E}}_{t\sim\mathcal{N}(1,\sigma^2)}\left[\mathbf{1}\{|t|\le 1\}\left(1-\frac{\mathcal{N}(\mu,\sigma^2;t)}{\mathcal{N}(1,\sigma^2;t)}\right)\right].\end{aligned}$$

We have that since $\mu > 1$ the ratio $\frac{\mathcal{N}(\mu,\sigma^2;t)}{\mathcal{N}(1,\sigma^2;t)} = e^{\frac{(\mu-1)(-\mu+2t-1)}{2\sigma^2}}$ is maximized for $t = 1$ and has maximum value $e^{-\frac{(\mu-1)^2}{2\sigma^2}}$. By taking the derivative with respect to σ we observe that the probability that $N(1,\sigma)$ assigns to $[-1,1]$ is decreasing with respect to σ and therefore it is minimized for $\sigma = 1$. We have that $\mathbf{Pr}_{t\sim\mathcal{N}(1,\sigma)}[-1 < t < 1] = \Omega(1/\sigma)$ and therefore $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}\left[-1 \le \boldsymbol{x}_i \le 1\right] - \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[-1 \le \boldsymbol{x}_i \le 1\right] \ge C \cdot \left(1 - e^{-\frac{(\mu-1)^2}{2\sigma^2}}\right)$. We can obtain the significantly weaker lower bound of $c\min(1,(1-|\mu|)^2)$ for some absolute constant $c \in (0,1]$ by using the inequality $1 - e^{-x} \ge 1/2\min(1,x)$ that holds for all $x \in [0,+\infty)$.

We now deal with the case $\mu \in [0,1)$. In that case the expression of their ratio of the densities of $\mathcal{N}(1,\sigma)$ and $\mathcal{N}(\mu,\sigma)$ derived above shows us that they cross at $t = (1+\mu)/2$. Therefore, they completely cancel out in the interval $[\mu, 1]$. We have $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}[-1 \le \boldsymbol{x}_i \le 1] - \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}[-1 \le \boldsymbol{x}_i \le 1] = \mathbf{Pr}_{t\sim\mathcal{N}(\mu,\sigma)}[-1 \le t \le \mu] - \mathbf{Pr}_{t\sim\mathcal{N}(1,\sigma)}[-1 \le t \le \mu] = \Omega((1-\mu)/(1+\sigma^4))$, where to obtain the last inequality we use standard approximations of Gaussian integrals. Combining the above two cases we obtain the claimed lower bound.

□

The Proof of Theorem 4.2.2

We are now ready to provide the complete proof of Theorem 4.2.2. Consider an instance of Max-Cut with $|V| = d$ and optimal value $\mathrm{opt} = O(d^2)$. Let $\boldsymbol{L}_G$ be the Laplacian matrix of the (connected) graph G. Since the minimum eigenvalue of $\boldsymbol{L}_G$ is 0, we project the matrix onto the subspace V that is orthogonal to $\mathbf{1} = (1,\dots,1)$. We introduce a $(d-1)\times d$ partial isometry $\boldsymbol{R}$, that satisfies $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$ and $\boldsymbol{R}\mathbf{1} = \mathbf{0}$, i.e., $\boldsymbol{R}$ projects vectors to the subspace V. We consider $\boldsymbol{L}'_G = \boldsymbol{R}\boldsymbol{L}_G\boldsymbol{R}^T$. It suffices to find a solution $\boldsymbol{x} \in V$ and then project back to $\mathbb{R}^d$: $\boldsymbol{y} = \boldsymbol{R}^T\boldsymbol{x}$. We note that the matrix $\boldsymbol{L}'_G$ is positive definite (the smallest eigenvalue of $\boldsymbol{L}'_G$ is equal to the second smallest eigenvalue of $\boldsymbol{L}_G$) and preserves the optimal score value, in the sense that

$$\mathrm{opt} = \max_{\boldsymbol{y}\in\mathbb{R}^d} \boldsymbol{y}^T\boldsymbol{L}_G\boldsymbol{y} = \max_{\boldsymbol{x}\in\mathbb{R}^d} (\boldsymbol{R}^T\boldsymbol{x})^T\boldsymbol{L}_G(\boldsymbol{R}^T\boldsymbol{x}) = \max_{\boldsymbol{x}\in V} \boldsymbol{x}^T\boldsymbol{L}'_G\boldsymbol{x}\,.$$

Assume that there exists an efficient black-box algorithm $\mathcal{A}$, that, given sample access to a generative process of coarse Gaussian data $\mathcal{N}_\pi(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})$ with known covariance [6] matrix $\boldsymbol{\Sigma}$, computes an estimate $\widetilde{\boldsymbol{\mu}}$ in $\mathrm{poly}(d)$ time, that satisfies

$$d_{\mathrm{TV}}(\mathcal{N}_\pi(\widetilde{\boldsymbol{\mu}},\boldsymbol{\Sigma}),\mathcal{N}_\pi(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})) < 1/d^c\,.$$

We choose the known covariance matrix to be equal to $\boldsymbol{\Sigma} = (\boldsymbol{L}'_G)^{-1}\mathrm{opt}$, where opt is the given optimal Max-Cut value and let $\boldsymbol{\mu}^\star \in \{-1,1\}^{d-1}$ be the unknown mean

[6] We remark that our hardness result is stated for identity covariance matrix (and not for an arbitrary known covariance matrix). In order to handle this case, we provide a detailed discussion after the end of the proof of Theorem 4.2.2.

vector. Recall that, not only the black-box algorithm $\mathcal{A}$, but also the generative process that we design is agnostic to the true mean. However, as we will see the knowledge of the optimal value opt and the fact that the true mean lies in the hypercube $\{-1, 1\}^{d-1}$ suffice to generate samples from the true coarse generative process $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$.

In what follows, we will construct such a coarse generative process using the objective function and the constraints of the Max-Cut problem. Specifically, we will design a collection $\mathcal{B} = \{\mathcal{S}_1, \dots, \mathcal{S}_{d-1}, \mathcal{T}\}$ of d partitions of the d-dimensional Euclidean space and let the partition distribution π be the uniform probability measure over $\mathcal{B}$.

We define the partitions as follows: for any $i = 1, \dots, d-1$, let $S_i = \{\boldsymbol{x} : -1 \leq \boldsymbol{x}_i \leq 1\}$ and $\mathcal{S}_i = \{S_i, S_i^c\}$. These $d-1$ partitions simulate the integrality constraints of Max-Cut , i.e., the solution vector should lie in the hypercube $\{-1, 1\}^{d-1}$. It remains to construct $\mathcal{T}$, which corresponds to the quadratic objective of Max-Cut . We let $T = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \leq q\}$, for $q > 0$ to be decided. Then, we let $\mathcal{T} = \{T, T^c\}$. Recall that the known covariance matrix $\boldsymbol{\Sigma} = (\boldsymbol{L}'_G)^{-1}$opt lies in $\mathbb{R}^{(d-1)\times(d-1)}$ and, so, we will use $d-1$ bands (i.e., fat hyperplanes).

The main question to resolve is how to generate efficiently samples from the designed general partition, i.e., the distribution $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$, *without* knowing the value of $\boldsymbol{\mu}^\star$. The key observation is that, by the rotation invariance of the Gaussian distribution, the probability $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; T) = \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})}\left[\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \leq q\right]$ is a constant p that only depends on the value opt of the maximum cut (see the proof of Lemma 4.2.5). Therefore, having this value p, we can flip a coin with this probability and give the coarse sample T if we get heads and T^c otherwise. At the same time, the value of $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$ is an absolute constant that does not depend on $\boldsymbol{\mu}^\star \in \{-1, 1\}^{d-1}$ and, therefore, we can again simulate coarse samples by flipping a coin with probability equal to $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$. More precisely, since S_i is a symmetric interval around 0, we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})}\left[-1 \leq \boldsymbol{x}_i \leq 1\right] = \mathop{\mathbf{Pr}}_{t\sim\mathcal{N}(1, \boldsymbol{\Sigma}_{ii})}\left[-1 \leq t \leq 1\right].$$

Notice that the above constant only depends on the *known* constant $\boldsymbol{\Sigma}_{ii}$ and can be computed to very high accuracy using well known approximations of the Gaussian integral or rejection sampling.

Moreover, all the probabilities $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i), \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; T)$ are at least polynomially small in $1/d$. In particular, $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$, is always larger than $\Omega(1/\sigma) \geq \mathrm{poly}(1/d)$ and smaller than $1/2$ and $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; T) = \Phi(1) + o(1/\sqrt{d})$ [7], see the proof of Lemma 4.2.5. Having these values we can generate samples from $\mathcal{N}_\pi$ as follows:

1. Pick one of the d sets $S_1, \dots, S_{d-1}, T$ uniformly at random.

2. Flip a coin with success probability equal to the probability of the corresponding sets and return either the set or its complement.

[7] $\Phi(\cdot)$ is the CDF of the standard Normal distribution.

Giving sample access to the designed oracle with $\mathcal{B} = \{\mathcal{S}_1, \ldots, \mathcal{S}_{d-1}, \mathcal{T}\}$, the black-box algorithm $\mathcal{A}$ computes efficiently and returns an estimate $\widetilde{\boldsymbol{\mu}} \in \mathbb{R}^{d-1}$, that satisfies

$$d_{\mathrm{TV}}(\mathcal{N}_\pi(\widetilde{\boldsymbol{\mu}}, \boldsymbol{\Sigma}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})) < o(1/d^c)\,.$$

We proceed with two claims: (i) the algorithm's output $\widetilde{\boldsymbol{\mu}}$ should lie in a ball of radius $\mathrm{poly}(1/d)$, centered at one of the vertices of the hypercube $\{-1,1\}^{d-1}$ and (ii) it will hold that the rounded vector $\widehat{\boldsymbol{\mu}} = (\mathrm{sgn}(\widetilde{\boldsymbol{\mu}}_i))_{1 \leq i \leq d-1} \in \{-1,1\}^{d-1}$ will attain a cut score, that approximates the Max-Cut within a factor larger than 16/17. By the algorithm's guarantee, since π is the uniform distribution, we get that

$$|\mathcal{N}(\widetilde{\boldsymbol{\mu}}, \boldsymbol{\Sigma}; T) - \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; T)| + \sum_{i=1}^{d-1} |\mathcal{N}(\widetilde{\boldsymbol{\mu}}, \boldsymbol{\Sigma}; S_i) - \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)| = o(1/d^{c-1})\,.$$

Hence, we get that each of the above d summands is at most $o(1/d^{c-1})$.

Claim 4. *It holds that* $\|\widetilde{\boldsymbol{\mu}} - \widehat{\boldsymbol{\mu}}\|_\infty < \epsilon$, *where* $\widetilde{\boldsymbol{\mu}}$ *is the black-box algorithm's estimate and* $\widehat{\boldsymbol{\mu}}$ *its rounding to* $\{-1,1\}^{d-1}$.

Proof. For any coordinate $i \in [d-1]$, we will apply Lemma 4.2.7 in order to bound the distance between the estimated guess and the true, based on the Gaussian mass gap in each one of the $d-1$ bands.

Note that $|\boldsymbol{\mu}_i^\star| = 1$ for all $i \in [d-1]$. Also, note that the $(d-1)\times(d-1)$ matrix $\boldsymbol{L}_G'$ is positive definite and the minimum eigenvalue $\lambda(\boldsymbol{L}_G')$ is equal to the second smallest eigenvalue of the $d \times d$ Laplacian matrix $\boldsymbol{L}_G$. It holds that $\lambda(\boldsymbol{L}_G') > 0$. Hence, the maximum entry of the covariance matrix $\boldsymbol{\Sigma} = (\boldsymbol{L}_G')^{-1}\mathrm{opt}$ is upper bounded by $1/(\mathrm{opt} \cdot \lambda(\boldsymbol{L}_G')) < Q = \mathrm{poly}(d)$ for some value Q. Using Lemma 4.2.7 and the algorithm's guarantee, we have that

$$(|\widetilde{\boldsymbol{\mu}}_i| - 1)^2/Q^4 \leq |\mathcal{N}(\widetilde{\boldsymbol{\mu}}, \boldsymbol{\Sigma}; S_i) - \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)| = o\left(1/d^{c-1}\right)\,.$$

For sufficiently large c, we get that each coordinate of the estimated vector $\widetilde{\boldsymbol{\mu}}$ lies in an interval, centered at either 1 or -1 of length $o(1/d^{c-1})$. This implies that $\|\widetilde{\boldsymbol{\mu}} - \boldsymbol{w}\|_\infty < \epsilon$ for some $\epsilon = o(1/d^{c-1})$ and some vertex $\boldsymbol{w}$ of the hypercube $\{-1,1\}^{d-1}$. Hence, we have that $\widetilde{\boldsymbol{\mu}}$ should lie in a ball, with respect to the L_∞ norm, centered at one of the vertices of the $(d-1)$-hypercube with radius of order ϵ and note that this vertex corresponds to the rounded vector $\widehat{\boldsymbol{\mu}}$ of the estimated vector. □

We continue by claiming that the rounded vector $\widehat{\boldsymbol{\mu}}$ attains a Max-Cut value, that approximates the optimal value opt withing a factor strictly larger than 16/17.

Claim 5. *The* Max-Cut *value of the rounded vector* $\widehat{\boldsymbol{\mu}} \in \{-1,1\}^{d-1}$ *satisfies*

$$\widehat{\boldsymbol{\mu}}^T \boldsymbol{L}_G' \widehat{\boldsymbol{\mu}} > (16/17) \cdot \mathrm{opt}\,.$$

Proof. We will make use of Lemma 4.2.5, in order to get the desired result via the Gaussian mass gap between the two means on the designed ellipsoid. In order to apply this Lemma, note that, for the true mean $\boldsymbol{\mu}^\star$, we have that $\|\boldsymbol{v}^\star\|_2^2 = \|(\boldsymbol{\Sigma}^\star)^{-1/2}\boldsymbol{\mu}^\star\|_2^2 = ((\boldsymbol{\mu}^\star)^T \boldsymbol{L}'_G \boldsymbol{\mu}^\star)/\mathrm{opt} = 1$, since the true mean attains the optimal Max-Cut score. Similarly, for the rounded estimated mean $\widehat{\boldsymbol{\mu}}$, the associated vector $\widehat{\boldsymbol{v}}$ satisfies $\|\widehat{\boldsymbol{v}}\|_2 \leq 1$, since its cut value is at most opt. So, we can apply Lemma 4.2.5 with $\boldsymbol{v}^\star = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}^\star$ and $\boldsymbol{v} = \boldsymbol{\Sigma}^{-1/2}\widehat{\boldsymbol{\mu}}$ and get that

$$\frac{1 - \left(\widehat{\boldsymbol{\mu}}^T \boldsymbol{L}'_G \widehat{\boldsymbol{\mu}}\right)/\mathrm{opt}}{6\sqrt{2d+4}} - o\left(1/\sqrt{d}\right) < o\left(1/d^{c-1}\right),$$

which implies that, for some small constant c', the value of the estimated mean satisfies $\widehat{\boldsymbol{\mu}}^T \boldsymbol{L}'_G \widehat{\boldsymbol{\mu}} > (1 - c' - 1/d^{c-1})\mathrm{opt}$. This implies that the algorithm $\mathcal{A}$ can approximate the Max-Cut value within a factor higher than 16/17. □

Known Covariance vs. Identity Covariance. Recall that our hardness result (Theorem 4.2.2) states that there is no algorithm with sample access to $\mathcal{N}_\pi(\boldsymbol{\mu}^\star) = \mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{I})$, that can compute a mean $\widetilde{\boldsymbol{\mu}} \in \mathbb{R}^d$ in poly(d) time such that $d_{\mathrm{TV}}(\mathcal{N}_\pi(\widetilde{\boldsymbol{\mu}}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star)) < 1/d^c$ for some absolute constant $c > 1$. In order to prove our hardness result, we assume that there exists such a black-box algorithm $\mathcal{A}$. Hence, to make use of $\mathcal{A}$, one should provide samples generated by a coarse Gaussian with *identity* covariance matrix. However, in our reduction, we show that we can generate samples from a coarse Gaussian (which is associated with the Max-Cut instance) that has *known* covariance matrix $\boldsymbol{\Sigma}$. Let us consider a sample $S \sim \mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$. Since $\boldsymbol{\Sigma}$ is known, we can rotate the sets and give as input to the algorithm $\mathcal{A}$ the set

$$\boldsymbol{\Sigma}^{-1/2} \cdot S := \left\{\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x} : \boldsymbol{x} \in S\right\},$$

i.e., we can implement the membership oracle $\mathcal{O}_{\boldsymbol{\Sigma}^{-1/2}\cdot S}(\cdot)$, assuming oracle access to $\mathcal{O}_S(\cdot)$. We have that $\mathcal{O}_{\boldsymbol{\Sigma}^{-1/2}\cdot S}(\boldsymbol{x}) = \mathcal{O}_S(\boldsymbol{\Sigma}^{1/2}\boldsymbol{x})$. We continue with a couple of observations.

1. We first observe that, for any partition $\mathcal{S}$ of the d-dimensional Euclidean space, there exists another partition $\boldsymbol{\Sigma}^{-1/2}\cdot\mathcal{S}$ consisting of the sets $\boldsymbol{\Sigma}^{-1/2}\cdot S$, where $S \in \mathcal{S}$. Note that since $\boldsymbol{\Sigma}^{-1/2}$ is full rank, the mapping $\boldsymbol{x} \mapsto \boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}$ is a bijection and so $\boldsymbol{\Sigma}^{-1/2}\cdot\mathcal{S}$ is a partition of the space with $\pi(\boldsymbol{\Sigma}^{-1/2}\cdot\mathcal{S}) = \pi(\mathcal{S})$.

2. We have that $\boldsymbol{x} \in S$ if and only if $\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x} \in \boldsymbol{\Sigma}^{-1/2}\cdot S$ and so

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}[\boldsymbol{1}\{\boldsymbol{x} \in S\}] = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}[\boldsymbol{1}\{\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x} \in \boldsymbol{\Sigma}^{-1/2}\cdot S\}].$$

Since it holds that $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ if and only if $\boldsymbol{w} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{z} + \boldsymbol{\mu}$ with $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, we get for an arbitrary subset $S \subseteq \mathbb{R}^d$ that

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}[\mathbf{1}\{\boldsymbol{x} \in S\}] &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\mathbf{1}\left\{\boldsymbol{\Sigma}^{-1/2}\left(\boldsymbol{\Sigma}^{1/2}\boldsymbol{x} + \boldsymbol{\mu}\right) \in \boldsymbol{\Sigma}^{-1/2} \cdot S\right\}\right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu},\boldsymbol{I})}\left[\mathbf{1}\{\boldsymbol{x} \in \boldsymbol{\Sigma}^{-1/2} \cdot S\}\right].\end{aligned}$$

Let us consider a set $S \subseteq \mathbb{R}^d$ distributed as $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$. This set is the one that the algorithm with the known covariance matrix works with. We are now ready to combine the above two observations in order to understand what is the input to the identity covariance matrix algorithm. We have that

$$\begin{aligned}\mathop{\mathbf{Pr}}_{S\sim\mathcal{N}_\pi(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}[S] &= \sum_{\mathcal{S}} \mathbf{1}\{S \in \mathcal{S}\}\pi(\mathcal{S})\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S) \\ &= \sum_{\mathcal{S}} \mathbf{1}\{S \in \mathcal{S}\}\pi(\mathcal{S})\mathcal{N}(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}^\star, \boldsymbol{I}; \boldsymbol{\Sigma}^{-1/2} \cdot S) \\ &= \sum_{\boldsymbol{\Sigma}^{-1/2}\cdot\mathcal{S}} \mathbf{1}\{\boldsymbol{\Sigma}^{-1/2} \cdot S \in \boldsymbol{\Sigma}^{-1/2} \cdot \mathcal{S}\}\pi(\boldsymbol{\Sigma}^{-1/2} \cdot \mathcal{S})\mathcal{N}(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}^\star, \boldsymbol{I}; \boldsymbol{\Sigma}^{-1/2} \cdot S) \\ &= \mathop{\mathbf{Pr}}_{S'\sim\mathcal{N}_{\pi'}(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}^\star,\boldsymbol{I})}[S'],\end{aligned}$$

where the set S' is distributed as $\mathcal{N}_{\pi'}(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}^\star, \boldsymbol{I})$ where π' is the 'rotated' partition distribution supported on the rotated partitions $\boldsymbol{\Sigma}^{-1/2} \cdot \mathcal{S}$ for each $\mathcal{S}$ with $\pi(\mathcal{S}) > 0$. We remark that the second equation follows from the second observation and the third equation from the first one. Hence, the algorithm $\mathcal{A}$ (the one that works with identity matrix) obtains the rotated sets (i.e., membership oracles) $\boldsymbol{\Sigma}^{-1/2} \cdot S$ and the (unknown) target mean vector is $\boldsymbol{u} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}^\star$.

4.2.3 Efficient Mean Estimation under Convex Partitions

In this section, we formally state and prove Theorem 4.2.3: we provide an efficient algorithm for Gaussian mean estimation under *convex* partitions. The following definition of information preservation is very similar with the one given in Definition 4.1.2. The difference is that we only require from π to preserve the distances of Gaussians around the true Gaussian $\mathcal{N}(\boldsymbol{\mu}^\star)$ as opposed to the distance of any pair of Gaussians $\mathcal{N}(\boldsymbol{\mu}^\star)$: this is a somewhat more flexible assumption about the partition distribution π and the true Gaussian $\mathcal{N}(\boldsymbol{\mu}^*)$ as a pair.

Definition 4.2.8 (Information Preserving Partition Distribution for Gaussians)**.** *Let $\alpha \in [0, 1]$ and consider a d-dimensional Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^\star)$. We say that π is an α-information preserving partition distribution with respect to the true Gaussian $\mathcal{N}(\boldsymbol{\mu}^\star)$ if for any Gaussian distribution $\mathcal{N}(\boldsymbol{\mu})$, it holds that $d_{\mathrm{TV}}(\mathcal{N}_\pi(\boldsymbol{\mu}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star)) \geq \alpha \cdot d_{\mathrm{TV}}(\mathcal{N}(\boldsymbol{\mu}), \mathcal{N}(\boldsymbol{\mu}^\star))$.*

We refer to Section 4.2.4 for a geometric condition, under which a partition is α-information preserving. In particular, we prove that a partition is α-information preserving if, for any hyperplane, it holds that the mass of the cells of the partition that do not intersect with the hyperplane is at least α. This is true for most natural partitions, see e.g., the Voronoi diagram of Figure 2.1.

In this section, we discuss and establish the two structural lemmata required in order to prove Theorem 4.2.3. Our goal is to maximize the empirical log-likelihood objective

$$\mathcal{L}_N(\boldsymbol{\mu}) = \frac{1}{N}\sum_{i=1}^{N} \log \mathcal{N}(\boldsymbol{\mu}; S_i)\,, \tag{4.1}$$

where the N (convex) sets $S_1, \ldots, S_N$ are drawn from the coarse Gaussian generative process $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$. We first show that the above empirical likelihood is a concave objective with respect to $\boldsymbol{\mu} \in \mathbb{R}^d$. In the following lemma, we show that the log-probability of a convex set S, i.e., the function $\log \mathcal{N}(\boldsymbol{\mu}; S)$ is a concave function of the mean $\boldsymbol{\mu}$.

Lemma 4.2.9 (Concavity of Log-Likelihood)**.** *Let $S \subseteq \mathbb{R}^d$ be a convex set. The function $\log \mathcal{N}(\boldsymbol{\mu}; S)$ is concave with respect to the mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$.*

In order to prove that the Hessian matrix of this objective is negative semi-definite, we use a variant of the Brascamp-Lieb inequality. Having established the concavity of the empirical log-likelihood, we next have to bound the sample complexity of the empirical log-likelihood. We prove the following lemma.

Lemma 4.2.10 (Sample Complexity of Empirical Log-Likelihood)**.** *Let $\epsilon, \delta \in (0,1)$ and consider a generative process for coarse d-dimensional Gaussian data $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$ (see Definition 4.2.1). Also, assume that every $\mathcal{S} \in \operatorname{supp}(\pi)$ is a convex partition of the Euclidean space. Let $N = \widetilde{\Omega}(d/(\epsilon^2\alpha^2)\log(1/\delta))$. Consider the empirical log-likelihood objective*

$$\mathcal{L}_N(\boldsymbol{\mu}) = \frac{1}{N}\sum_{i=1}^{N} \log \mathcal{N}(\boldsymbol{\mu}; S_i)\,.$$

Then, with probability at least $1-\delta$, we have that, for any Gaussian distribution $\mathcal{N}(\boldsymbol{\mu})$ that satisfies $d_{\mathrm{TV}}(\mathcal{N}(\boldsymbol{\mu}), \mathcal{N}(\boldsymbol{\mu}^\star)) \geq \epsilon$, it holds that $\max_{\widetilde{\boldsymbol{\mu}} \in \mathbb{R}^d} \mathcal{L}_N(\widetilde{\boldsymbol{\mu}}) - \mathcal{L}_N(\boldsymbol{\mu}) \geq \Omega(\epsilon^2\alpha^2)\,.$

The above lemma states that, given roughly $\widetilde{O}(d/(\epsilon^2\alpha^2))$ samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, we can guarantee that the maximizer $\widetilde{\boldsymbol{\mu}}$ of the empirical log-likelihood achieves a total variation gap at most ϵ against the true mean vector $\boldsymbol{\mu}^\star$, i.e., $d_{\mathrm{TV}}(\mathcal{N}(\widetilde{\boldsymbol{\mu}}), \mathcal{N}(\boldsymbol{\mu}^\star)) \leq \epsilon$. In fact, thanks to the concavity of the empirical log-likelihood objective, it suffices to show that Gaussian distributions $\mathcal{N}(\boldsymbol{\mu})$, that satisfy $d_{\mathrm{TV}}(\mathcal{N}(\boldsymbol{\mu}), \mathcal{N}(\boldsymbol{\mu}^\star)) > \epsilon$, will also be significantly sub-optimal solutions of the empirical log-likelihood maximization. The key idea in order to attain the desired sample complexity, is that is suffices to focus on guess vectors $\boldsymbol{\mu}$ that lie in a sphere of radius $\Omega(\epsilon)$. Technically, the proof of Lemma 4.2.10 relies on a concentration result of likelihood ratios and in the observation that, while the empirical log-likelihood objective $\mathcal{L}_N$ is concave (under convex partitions), the regularized objective $\mathcal{L}_N(\boldsymbol{\mu}) + \|\boldsymbol{\mu}\|_2^2$ is convex with respect to the guess mean vector $\boldsymbol{\mu}$.

Concavity of Log-likelihood: Proof of Lemma 4.2.9

In this section, we show that the log-likelihood is concave when the underlying partitions are convex. The Hessian of the log-likelihood $\mathcal{L}$ for the set S has a notable property. When restricted to a direction $\boldsymbol{v} \in \mathbb{R}^d$, the quadratic $\boldsymbol{v}^T(\nabla^2 \mathcal{L})\boldsymbol{v}$ quantifies the variance reduction, observed between the distributions $\mathcal{N}_S$ (Gaussian conditioned on S) and $\mathcal{N}$ (unrestricted Gaussian, i.e., $S = \mathbb{R}^d$). When the set S is convex (and, hence the indicator function $\mathbf{1}_S$ is log-concave), the variance of the unrestricted Gaussian is always larger than the conditional one. This intriguing result is an application of a variation of the Brascamp-Lieb inequality, due to Hargé (see Lemma 4.2.11 for the inequality that we utilize). Recall that, both the empirical and the population log-likelihood objectives are convex combinations of the function $f(\boldsymbol{\mu}, \boldsymbol{\Sigma}; S) = \log \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; S)$ and, hence, it suffices to show that f is concave with respect to $\boldsymbol{\mu} \in \mathbb{R}^d$, when the set S is convex.

Proof of Lemma 4.2.9. Without loss of generality, we can take $\boldsymbol{\Sigma} = \boldsymbol{I} \in \mathbb{R}^{d \times d}$. Let $f(\boldsymbol{\mu}; S) = \log \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; S)$ for an arbitrary convex set $S \subseteq \mathbb{R}^d$. The gradient $\nabla_{\boldsymbol{\mu}} f(\boldsymbol{\mu})$ of f with respect to $\boldsymbol{\mu}$ is equal to

$$\nabla_{\boldsymbol{\mu}} \left(\log \int_S \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{x}-\boldsymbol{\mu})}{2} \right) d\boldsymbol{x} \right) = \frac{\int_S \boldsymbol{x} \exp(-(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{x}-\boldsymbol{\mu})/2) d\boldsymbol{x}}{\int_S \exp(-(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{x}-\boldsymbol{\mu})/2) d\boldsymbol{x}} - \boldsymbol{\mu} \,.$$

Hence, we get that

$$\nabla_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}[\boldsymbol{x}] - \boldsymbol{\mu} \,.$$

We continue with the computation of the Hessian of the function f with respect to $\boldsymbol{\mu}$

$$\nabla^2_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) = -\boldsymbol{I} + \frac{\int_S \boldsymbol{x}(\boldsymbol{x}-\boldsymbol{\mu})^T \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; \boldsymbol{x}) d\boldsymbol{x}}{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; S)} - \frac{\left(\int_S \boldsymbol{x} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; \boldsymbol{x}) d\boldsymbol{x} \right) \left(\int_S (\boldsymbol{x}-\boldsymbol{\mu})^T \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; \boldsymbol{x}) d\boldsymbol{x} \right)}{\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; S)^2} \,,$$

and, so, we have that

$$\nabla^2_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) = -\boldsymbol{I} + \left(\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}[\boldsymbol{x}\boldsymbol{x}^T] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}[\boldsymbol{x}] \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}[\boldsymbol{x}^T] \right) = \mathop{\mathbf{Cov}}_{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}[\boldsymbol{x}] - \boldsymbol{I} \,.$$

Observe that, when $S = \mathbb{R}^d$, we get that both the gradient and the Hessian vanish. In order to show the concavity of f with respect to the mean vector $\boldsymbol{\mu}$, consider an arbitrary vector $\boldsymbol{v} \in \mathbb{R}^d$ in the ball $\|\boldsymbol{v}\|_2 = 1$. We have the quadratic form

$$\boldsymbol{v}^T \nabla^2_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) \boldsymbol{v} = \boldsymbol{v}^T \underset{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}{\mathbf{Cov}} [\boldsymbol{x}] \boldsymbol{v} - 1 = \underset{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}{\mathbb{E}} \left[(\boldsymbol{v}^T \boldsymbol{x})^2 \right] - \left(\underset{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}{\mathbb{E}} [\boldsymbol{v}^T \boldsymbol{x}] \right)^2 - 1 \,.$$

In order to show the desired inequality, we will apply the following variant of the Brascamp-Lieb inequality.

Lemma 4.2.11 (Brascamp-Lieb Inequality, Hargé (see (Gui09)))**.** *Let g be convex function on $\mathbb{R}^d$ and let S be a convex set on $\mathbb{R}^d$. Let $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be the Gaussian distribution on $\mathbb{R}^d$. It holds that*

$$\underset{\boldsymbol{x} \sim \mathcal{N}_S}{\mathbb{E}} \left[g \left(\boldsymbol{x} + \boldsymbol{\mu} - \underset{\boldsymbol{x} \sim \mathcal{N}_S}{\mathbb{E}} [\boldsymbol{x}] \right) \right] \le \underset{\boldsymbol{x} \sim \mathcal{N}}{\mathbb{E}} [g(\boldsymbol{x})] \,. \tag{4.2}$$

We apply the above Lemma with $g(\boldsymbol{x}) = (\boldsymbol{v}^T \boldsymbol{x})^2$. We get that

$$\int_{\mathbb{R}^d} (\boldsymbol{v}^T (\boldsymbol{x} + \boldsymbol{\mu} - \underset{y \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}{\mathbb{E}} \boldsymbol{y}))^2 \cdot \frac{\mathbf{1}_S(\boldsymbol{x}) \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; \boldsymbol{x}) d\boldsymbol{x}}{\int_{\mathbb{R}^d} \mathbf{1}_S(\boldsymbol{x}) \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; \boldsymbol{x}) d\boldsymbol{x}} \le \int_{\mathbb{R}^d} (\boldsymbol{v}^T \boldsymbol{x})^2 \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; \boldsymbol{x}) d\boldsymbol{x} \,.$$

Hence, we get the desired variance reduction in the direction $\boldsymbol{v}$

$$\underset{\boldsymbol{x} \sim \mathcal{N}_S(\boldsymbol{\mu}, \boldsymbol{I})}{\mathrm{Var}} [\boldsymbol{v}^T \boldsymbol{x}] \le \underset{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I})}{\mathrm{Var}} [\boldsymbol{v}^T \boldsymbol{x}] \,,$$

that implies the concavity of the function $\log \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; S)$ for convex sets S with respect to the mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$. □

Sample Complexity of Empirical Log-Likelihood: Proof of Lemma 4.2.10

In this section, we provide the proof of Lemma 4.2.10. This lemma analyzes the sample complexity of the empirical log-likelihood maximization $\mathcal{L}_N$, whose concavity (in convex partitions) was established in Lemma 4.2.9. We show that, given roughly $N = \widetilde{O}(d/(\epsilon^2 \alpha^2))$ samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, we can guarantee that Gaussian distributions $\mathcal{N}(\boldsymbol{\mu})$ with mean vectors $\boldsymbol{\mu}$, that are far from the true Gaussian $\mathcal{N}(\boldsymbol{\mu}^\star)$ in total variation distance, will also be sub-optimal solutions of the empirical maximization of the log-likelihood objective, i.e., they are far from being maximizers of the empirical log-likelihood objective. We first give an overview of the proof of Lemma 4.2.10. In Proposition 4.1.5 we provided a similar sample complexity bound for an empirical log-likelihood objective. However, in contrast to the analysis of Proposition 4.1.5, the parameter space is now unbounded – $\boldsymbol{\mu}$ can be any vector of $\mathbb{R}^d$ – and we cannot construct a cover of the whole space with finite size. However, thanks to the concavity of the empirical log-likelihood objective $\mathcal{L}_N$, we can show that it suffices to focus on guess vectors $\boldsymbol{\mu}$ that lie in a sphere $\partial \mathcal{B}$ (i.e., the boundary of a ball $\mathcal{B}$) of radius $\Omega(\epsilon)$. This argument heavily relies on the claim that the maximizer of the empirical log-likelihood $\mathcal{L}_N$ lies inside $\mathcal{B}$, which can be verified by monotonicity properties of the log-likelihood.

Afterwards, we consider a discretization $\mathcal{C}$ of the sphere and, for any vector $\boldsymbol{\mu} \in \mathcal{C}$, we can prove that $\mathcal{L}_N(\boldsymbol{\mu}^\star) - \mathcal{L}_N(\boldsymbol{\mu}) \geq \Omega(\alpha^2\epsilon^2)$. The main technical tool for this claim is a concentration result on likelihood ratios and the fact that the partition distribution is α-information preserving. In order to extend this property to the whole sphere, we exploit the convexity (with respect to $\boldsymbol{\mu}$) of a regularized version of the empirical log-likelihood objective $\mathcal{L}_N(\boldsymbol{\mu}) + \|\boldsymbol{\mu}\|_2^2$. The complete proof follows.

Proof of Lemma 4.2.10. Let $\widetilde{\boldsymbol{\mu}}$ be the maximizer of the empirical log-likelihood objective

$$\widetilde{\boldsymbol{\mu}} = \arg\max_{\boldsymbol{\mu} \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \log \mathcal{N}(\boldsymbol{\mu}; S_i)\,.$$

Since $\widetilde{\boldsymbol{\mu}}$ is the maximizer of the empirical objective, it is sufficient to prove that for any Gaussian $\mathcal{N}(\boldsymbol{\mu})$ whose total variation distance with $\mathcal{N}(\boldsymbol{\mu}^\star)$ is greater than ϵ, it holds that $\mathcal{L}_N(\boldsymbol{\mu}^\star) - \mathcal{L}_N(\boldsymbol{\mu}) \geq \Omega(\alpha^2\epsilon^2)$.

Moreover, we know that when $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2$ is smaller than some sufficiently small absolute constant, it holds $d_{\mathrm{TV}}(\mathcal{N}(\boldsymbol{\mu}_1), \mathcal{N}(\boldsymbol{\mu}_2)) \geq \Omega(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2)$. Therefore, any Gaussian whose mean $\boldsymbol{\mu}$ is far from $\boldsymbol{\mu}^\star$, i.e., $\|\boldsymbol{\mu} - \boldsymbol{\mu}^\star\|_2 \geq \Omega(\epsilon)$ will be in total variation distance at least ϵ from $\mathcal{N}(\boldsymbol{\mu}^\star)$ Therefore, to prove the lemma, it suffices to prove it for Gaussians whose means lie outside of a ball $\mathcal{B}$ of radius $\rho := \Omega(\epsilon)$ around $\boldsymbol{\mu}^\star$.

Since all observed sets S_i are convex, the empirical log-likelihood objective $\mathcal{L}_N(\boldsymbol{\mu})$ is concave with respect to $\boldsymbol{\mu}$, see Lemma 4.2.9. Since $\mathcal{L}_N$ is concave, it suffices to prove that for any $\boldsymbol{\mu}$ that lies exactly on the sphere of radius ρ, i.e., the surface of the ball $\mathcal{B}$ it holds $\mathcal{L}_N(\boldsymbol{\mu}^\star) - \mathcal{L}_N(\boldsymbol{\mu}) \geq \Omega(\alpha^2\epsilon^2)$. To prove this we first show that the maximizer of the empirical objective $\widetilde{\boldsymbol{\mu}}$ has to lie inside the ball $\mathcal{B}$. Assuming that $\widetilde{\boldsymbol{\mu}}$ lies outside of $\mathcal{B}$, let $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ be the antipodal points on the sphere $\partial\mathcal{B}$ that belong to the line $\widetilde{\boldsymbol{\mu}}$ connecting $\widetilde{\boldsymbol{\mu}}$ and $\boldsymbol{\mu}^\star$ and assume that $\boldsymbol{r}_2$ lies between $\boldsymbol{\mu}^\star$ and $\widetilde{\boldsymbol{\mu}}$. In that case the restriction of $\mathcal{L}_N$ on that line cannot be concave, since it has to be increasing from $\boldsymbol{r}_1$ to $\boldsymbol{\mu}^\star$, decreasing from $\boldsymbol{\mu}^\star$ to $\boldsymbol{r}_2$ and then increase again from $\boldsymbol{r}_2$ to $\widetilde{\boldsymbol{\mu}}$. Thus, $\widetilde{\boldsymbol{\mu}}$ lies inside $\mathcal{B}$. Now, by concavity of $\mathcal{L}_N$, we obtain that, by projecting any point $\boldsymbol{\mu}$ that lies outside of the ball $\mathcal{B}$ onto $\mathcal{B}$, we can only increase its empirical likelihood. Therefore, it suffices to consider only points that lie on the sphere $\partial\mathcal{B}$.

We will now show that the claim is true for any $\boldsymbol{\mu} \in \partial\mathcal{B}$. We can create a cover of the sphere of radius $\rho\sqrt{1 + c\alpha^2}$, centered at $\boldsymbol{\mu}^\star$ for some sufficiently small absolute constant $c > 0$, whose convex hull contains $\mathcal{B}$. The following lemma shows that such a cover can be constructed with $(1/(\alpha\epsilon))^{O(d)}$ points.

Lemma 4.2.12 (see, e.g., Corollary 4.2.13 of (Ver18)). *For any $\epsilon > 0$, there exists an ϵ-cover $\mathcal{C}$ of the unit sphere in $\mathbb{R}^k$, with respect to the ℓ_2-norm, of size $O((1/\epsilon)^k)$. Moreover, the convex hull of the cover $\mathcal{C}$ contains the sphere of radius $1 - \epsilon$.*

Since the partition distribution π is α-information preserving we obtain that for any $\boldsymbol{\mu} \in \mathcal{C}$, it holds $d_{\mathrm{TV}}(\mathcal{N}_\pi(\boldsymbol{\mu}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star)) \geq \Omega(\alpha\epsilon)$. Applying Lemma 4.1.6 with $x = O(\log(|\mathcal{C}|/\delta)) = O(d \log(1/(\epsilon\delta)))$, we get that, with $N = \widetilde{O}(d/(\alpha^2\epsilon^2)\log(1/\delta))$, with probability at least $1-\delta$, it holds that, for any $\boldsymbol{\mu}$ in the cover $\mathcal{C}$, we have

$$\mathcal{L}_N(\boldsymbol{\mu}^\star) - \mathcal{L}_N(\boldsymbol{\mu}) \geq d_{\mathrm{TV}}(\mathcal{N}_\pi(\boldsymbol{\mu}^\star), \mathcal{N}_\pi(\boldsymbol{\mu}))^2 - \alpha^2\epsilon^2/2 \geq \Omega(\alpha^2\epsilon^2)\,. \tag{4.3}$$

Next, we need to extend this bound from the elements of the cover $\mathcal{C}$ to all elements of the sphere $\partial\mathcal{B}$. In what follows, in order to simplify notation, we may assume without loss of generality that $\boldsymbol{\mu}^\star = \mathbf{0}$. We are going to use the fact that $\log(\mathcal{N}(\boldsymbol{\mu}; S_i)) + \|\boldsymbol{\mu}\|_2^2/2$ is convex. To see that, write

$$\log(\mathcal{N}(\boldsymbol{\mu}; S_i)) + \|\boldsymbol{\mu}\|_2^2/2 = \log\left(e^{\|\boldsymbol{\mu}\|_2^2/2}\int_S e^{-\|\boldsymbol{x}-\boldsymbol{\mu}\|_2^2/2} d\boldsymbol{x}\right) = \log\left(\int_S e^{-\|\boldsymbol{x}\|_2^2/2 + \boldsymbol{x}^T\boldsymbol{\mu}} d\boldsymbol{x}\right),$$

which is a log-sum-exp function and thus convex (this can also be verified by directly computing the Hessian with respect to $\boldsymbol{\mu}$). This means that $\mathcal{L}_N(\boldsymbol{\mu}) + \|\boldsymbol{\mu}\|_2^2$ is also convex with respect to $\boldsymbol{\mu}$. Let $\boldsymbol{\mu} \in \partial B$. From the construction of the cover $\mathcal{C}$, we have that its convex hull contains the sphere ∂B. Therefore, $\boldsymbol{\mu}$ can be written as a convex combination of points of the cover, i.e., $\boldsymbol{\mu} = \sum_{i=1}^{|\mathcal{C}|} \alpha_i \boldsymbol{\mu}_i$, where $\boldsymbol{\mu}_i \in \mathcal{C}$. The convexity of $\mathcal{L}_N(\boldsymbol{\mu}) + \|\boldsymbol{\mu}\|_2^2$ implies that

$$\mathcal{L}_N(\boldsymbol{\mu}) + \|\boldsymbol{\mu}\|_2^2 \leq \sum_{i=1}^{|\mathcal{C}|} \alpha_i(\mathcal{L}_N(\boldsymbol{\mu}_i) + \|\boldsymbol{\mu}_i\|_2^2) \leq \max_i \mathcal{L}_N(\boldsymbol{\mu}_i) + \rho^2(1 + c\alpha^2)\,,$$

where to get the last inequality we used the fact that all points of our cover $\mathcal{C}$ belong to the sphere of radius $\rho\sqrt{1+c\alpha^2}$. Since $\|\boldsymbol{\mu}\|_2^2 = \rho^2$ the above inequality implies that $\mathcal{L}_N(\boldsymbol{\mu}) \leq \max_i \mathcal{L}_N(\boldsymbol{\mu}_i) + c\alpha^2\rho^2$. Combining this inequality with Equation (4.3), we obtain that, since c is sufficiently small and $\rho = \Theta(\epsilon)$, it holds $\mathcal{L}_N(\boldsymbol{\mu}) \leq \mathcal{L}_N(\boldsymbol{\mu}^\star) - \Omega(\epsilon^2\alpha^2)$. □

The Proof of Theorem 4.2.3

We conclude this section with the proof of Theorem 4.2.3. Since the likelihood function is concave (and therefore can be efficiently optimized) we focus mainly on bounding the sample complexity of our algorithm.

Proof of Theorem 4.2.3. Let us assume that the partition distribution π is α-information preserving and that is supported on *convex partitions* of $\mathbb{R}^d$. Our goal is to show that there exists an algorithm, that draws $\widetilde{O}(d/(\epsilon^2\alpha^2)\log(1/\delta))$ samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$ and computes an estimate $\widetilde{\boldsymbol{\mu}} \in \mathbb{R}^d$ so that $d_{\mathrm{TV}}(\mathcal{N}(\widetilde{\boldsymbol{\mu}}), \mathcal{N}(\boldsymbol{\mu}^\star)) \leq \epsilon$ with probability at least $1-\delta$. The algorithm works as follows: it optimizes the empirical log-likelihood objective

$$\mathcal{L}_N(\boldsymbol{\mu}) = \frac{1}{N}\sum_{i=1}^{N} \log \mathcal{N}(\boldsymbol{\mu}; S_i)\,,$$

where the samples are i.i.d. and $S_i \sim \mathcal{N}_\pi(\boldsymbol{\mu}^\star)$ for any $i \in [N]$. Using Lemma 4.2.9, we establish that the function $\mathcal{L}_N$ is concave with respect to the mean $\boldsymbol{\mu} \in \mathbb{R}^d$. This follows from the fact that convex combinations of concave functions remain concave. From Lemma 4.2.10, we obtain that it suffices to compute a point $\boldsymbol{\mu}$ such that $\mathcal{L}_N(\boldsymbol{\mu}) \geq \max_{\boldsymbol{\mu}'} \mathcal{L}_N(\boldsymbol{\mu}') - O(\alpha^2\epsilon^2)$. Specifically, given roughly $\widetilde{O}(d/(\epsilon^2\alpha^2))$ samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, we can guarantee, with high probability, that the maximizer $\widetilde{\boldsymbol{\mu}}$ of the empirical log-likelihood achieves a total variation gap at most ϵ against the true mean vector $\boldsymbol{\mu}^\star$, i.e., $d_{\mathrm{TV}}(\mathcal{N}(\widetilde{\boldsymbol{\mu}}), \mathcal{N}(\boldsymbol{\mu}^\star)) \leq \epsilon$. □

We proceed with a discussion about the running time of the above algorithm. Since $\mathcal{L}_N(\boldsymbol{\mu})$ is a concave function with respect to $\boldsymbol{\mu}$, this can be done efficiently. For example, we may perform gradient-ascent: for a fixed convex set $S \subseteq \mathbb{R}^d$ the gradient of the function $f(\boldsymbol{\mu}) = \log \mathcal{N}(\boldsymbol{\mu}; S) = \log \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu})}\left[\mathbf{1}\{\boldsymbol{x} \in S\}\right]$ (see Lemma 4.2.9) is equal to

$$\nabla_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_S(\boldsymbol{\mu})}[\boldsymbol{x}] - \boldsymbol{\mu}\,.$$

In order to compute the gradient of f, it suffices to approximately compute $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_S(\boldsymbol{\mu})}[\boldsymbol{x}]$ $= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu})}[\boldsymbol{x}\ \mathbf{1}\{\boldsymbol{x} \in S\}] / \mathcal{N}(\boldsymbol{\mu}; S)$. Both terms of this ratio can be estimated using independent samples from the distribution $\mathcal{N}(\boldsymbol{\mu})$ and access to the oracle $\mathcal{O}_S(\cdot)$, since the mean $\boldsymbol{\mu}$ is known (the current guess of the learning algorithm). Hence, the running time will be polynomial in the number of samples using, e.g., the ellipsoid algorithm.

Remark 3. *We remark that a precise calculation of the runtime would also depend on the regularity of the concave objective (Lipschitz or smoothness assumptions etc.) which in turn depend on the geometric properties of the sets. We opt not to track such dependencies since our main result is that, in this setting, the likelihood objective is concave and therefore can be efficiently optimized using standard black-box optimization techniques.*

4.2.4 Geometric Information Preservation

In this section, we aim to provide some intuition behind the notion of information preserving partitions. The following result provides a geometric property for the partition distribution π. We show that if the partition distribution satisfies this particular geometric property, then it is also information preserving. We underline that the geometric property is quite important for our better understanding and it has the advantage that it is easy to verify. Hence, while the notion of information preserving distributions may be less intuitive, we believe that the geometric preservation property that we state in Lemma 4.2.13 can fulfill this lack of intuition. The property informally states that, for any hyperplane, the sets in the partition that are not cut by this hyperplane have non trivial probability mass with respect to

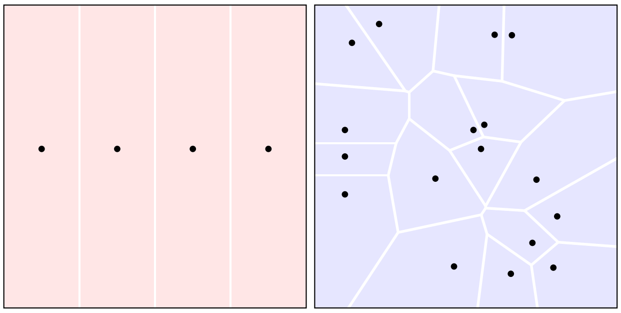


Figure 4.3: (a) is a very rough partition that makes learning the mean impossible: Gaussians $\mathcal{N}((0,z))$ centered along the same vertical line $(0,z)$ assign exactly the same probability to all cells of the partitions and therefore, $d_{\mathrm{TV}}(\mathcal{N}_\pi((0,z_1)), \mathcal{N}_\pi((0,z_2))) = 0$: it is impossible to learn the second coordinate of the mean. (b) is a convex partition of $\mathbb{R}^2$, that makes recovering the Gaussian possible.

the true Gaussian. In the case of mixtures of convex partitions, we would like the same property to hold in expectation.

Before stating Lemma 4.2.13, let us return to Figure 4.3. Observe that, in the first example with the four halfspaces, the geometric property does not hold, since there exists a line (i.e., a hyperplane) that intersects with all the sets. On the other hand, if we consider the second example with the Voronoi partition and assume that the true mean lies in the middle of the picture, we can see that any hyperplane does not intersect with a sufficient number of sets and, hence, the union of the uncut sets has non trivial probability mass for any hyperplane.

For a hyperplane $\mathcal{H}_{\boldsymbol{w},c} = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{w}^T\boldsymbol{x} = c\}$ with normal vector $\boldsymbol{w} \in \mathbb{R}^d$ and threshold $c \in \mathbb{R}$, we denote the two associated halfspaces by $\mathcal{H}^+_{\boldsymbol{w},c} = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{w}^T\boldsymbol{x} > c\}$ and $\mathcal{H}^-_{\boldsymbol{w},c} = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{w}^T\boldsymbol{x} < c\}$. Before stating the next Lemma, we shortly describe what means for a hyperplane to cut a set with respect to a Gaussian $\mathcal{N}$. The set S is not cut by the hyperplane $\mathcal{H}$, if it totally lies in a halfspace induced by the hyperplane, say $\mathcal{H}^+$, i.e., it holds that $\mathcal{N}(S) = \mathcal{N}(S \cap \mathcal{H}^+)$.

Lemma 4.2.13 (Geometric Information Preservation)**.** *Consider the generative process of coarse d-dimensional Gaussian data $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, (see Definition 4.2.1). Consider an arbitrary hyperplane $\mathcal{H}_{\boldsymbol{w},c}$ with normal vector $\boldsymbol{w} \in \mathbb{R}^d$ and threshold $c \in \mathbb{R}$. For a partition $\mathcal{S} \in \mathrm{supp}(\pi)$ of $\mathbb{R}^d$, consider the collection that contains all the sets that are not cut by the hyperplane $\mathcal{H}_{\boldsymbol{w},c}$, i.e.,*

$$U_{\boldsymbol{w},c,\mathcal{S}} = \bigcup \left\{ S \in \mathcal{S} : \mathcal{N}^\star(S \cap \mathcal{H}^+_{\boldsymbol{w},c}) = \mathcal{N}^\star(S) \vee \mathcal{N}^\star(S \cap \mathcal{H}^-_{\boldsymbol{w},c}) = \mathcal{N}^\star(S) \right\}.$$

Assume that π satisfies

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\pi} \left[\mathcal{N}(\boldsymbol{\mu}^\star; U_{\boldsymbol{w},c,\mathcal{S}}) \right] \geq \alpha\,, \tag{4.4}$$

for some $\alpha \in (0, 1]$. Then, for any Gaussian distribution $\mathcal{N}(\boldsymbol{\mu})$, it holds that

$$d_{\mathrm{TV}}(\mathcal{N}_\pi(\boldsymbol{\mu}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star)) \geq C_\alpha \cdot d_{\mathrm{TV}}(\mathcal{N}(\boldsymbol{\mu}), \mathcal{N}(\boldsymbol{\mu}^\star))\,,$$

for some C_α that depends only on α and satisfies $C_\alpha = \mathrm{poly}(\alpha)$, i.e., the partition distribution is C_α-information preserving.

Hence, the above geometric property is sufficient for information preservation. If we assume that the total variation distance between the true Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^\star)$ and a possible guess $\mathcal{N}(\boldsymbol{\mu})$ is at least ϵ and the partition distribution satisfies the geometric property of Equation (4.4), we get that the coarse generative process preserves a sufficiently large gap, in the sense that $d_{\mathrm{TV}}(\mathcal{N}_\pi(\boldsymbol{\mu}^\star), \mathcal{N}_\pi(\boldsymbol{\mu})) \geq \mathrm{poly}(\alpha)\epsilon$. The proof of the above lemma, which relies on high-dimensional anti-concentration results on Gaussian distributions, follows.

Proof of Lemma 4.2.13. Let us denote the true distribution by $\mathcal{N}^\star = \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{I})$ for short. Consider an arbitrary hyperplane $\mathcal{H}_{\boldsymbol{w},c}$ with normal vector $\boldsymbol{w} \in \mathbb{R}^d$ and threshold $c \in \mathbb{R}$. Since the partition distribution (supported on a family of partitions $\mathcal{B}$) satisfies Equation (4.4), we have that, for the random variable $\mathcal{N}^\star(U_{\boldsymbol{w},c,\mathcal{S}})$, that takes values in $[0, 1]$, there exists α such that

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\pi}\left[\mathcal{N}^\star(U_{\boldsymbol{w},c,\mathcal{S}})\right] = \alpha\,.$$

We will use the following simple Markov-type inequality for bounded random variables.

Fact 3 (Lemma B.1 from (SSBD14))**.** *Let Z be a random variable that takes values in $[0, 1]$. Then, for any $\alpha \in (0, 1)$, it holds that*

$$\mathbf{Pr}[Z > \alpha] \geq \frac{\mathbb{E}[Z] - \alpha}{1 - \alpha} \geq \mathbb{E}[Z] - \alpha\,.$$

By the Fact 3, it holds that

$$\mathop{\mathbf{Pr}}_{\mathcal{S}\sim\pi}\left[\mathcal{N}^\star(U_{\boldsymbol{w},c,\mathcal{S}}) \geq \alpha/2\right] \geq \alpha/2\,.$$

Hence, the mass of the “good” partitions is at least $\alpha/2$. Fix such a partition $\mathcal{S} \in \mathcal{B}$ (in the support of the partition distribution) and consider the true $\mathcal{N}^\star = \mathcal{N}(\boldsymbol{\mu}^\star)$ and the guess $\mathcal{N} = \mathcal{N}(\boldsymbol{\mu})$ distributions. For this pair of distributions, consider the set

$$\mathcal{H} = \left\{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x}^T(\boldsymbol{\mu} - \boldsymbol{\mu}^\star) = \left(\|\boldsymbol{\mu}\|_2^2 - \|\boldsymbol{\mu}^\star\|_2^2\right)/2\right\}.$$

Observe that this set is a hyperplane with normal vector $\boldsymbol{\mu}^\star - \boldsymbol{\mu}$, that contains the midpoint $\frac{1}{2}(\boldsymbol{\mu} + \boldsymbol{\mu}^\star)$ (see Figure 4.4).

Our main focus is to lower bound the total variation distance of the coarse distributions $\mathcal{N}_\pi^\star$ and $\mathcal{N}_\pi$. We claim that this lower bound can be described as a

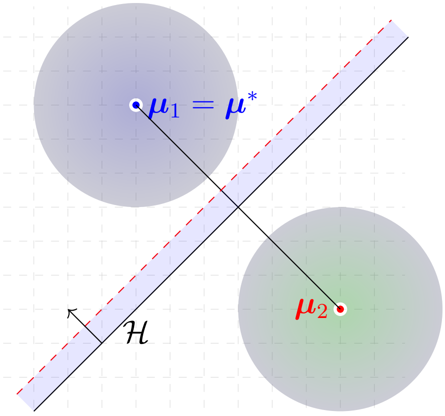


Figure 4.4: Illustration of the worst-case set in testing the hypotheses $h_1 = \{\boldsymbol{\mu}_1 = \boldsymbol{\mu}^\star\}$ and $h_2 = \{\boldsymbol{\mu}_2 = \boldsymbol{\mu}^\star\}$.

fractional knapsack problem and, hence, it is attained by a worst-case set, that (intuitively) places points as close as possible to the hyperplane $\mathcal{H}$, until its mass with respect to the true Gaussian $\mathcal{N}^\star$ is at least $\alpha/2$. Recall that the total variation distance between the two coarse distributions is

$$d_{\mathrm{TV}}(\mathcal{N}_\pi, \mathcal{N}_\pi^\star) = \sum_{\mathcal{S} \in \mathcal{B}} \pi(\mathcal{S}) \sum_{S \in \mathcal{S}} \Big| \mathcal{N}(S) - \mathcal{N}^\star(S) \Big| \,.$$

So, the LHS is at least $\Theta(\alpha)$ times the absolute gap of the masses assigned by $\mathcal{N}$ and $\mathcal{N}^\star$ over a worst-case set that lies in a good partition (one with $\mathcal{N}^\star(U_{\boldsymbol{w},c,\mathcal{S}}) \geq \alpha/2$). This holds since the probability to draw a good partition is at least $\alpha/2$. The following optimization problem gives a lower bound on the mass gap of a worst-case set in a good partition and, consequently, a lower bound on the total variation distance between $\mathcal{N}_\pi^\star$ and $\mathcal{N}_\pi$.

$$\min_S \left| \int (\mathcal{N}(\boldsymbol{\mu}^\star; \boldsymbol{x}) - \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{x})) \mathbf{1}_S(\boldsymbol{x}) d\boldsymbol{x} \right| ,$$

$$\text{subj. to} \quad \int \mathcal{N}(\boldsymbol{\mu}^\star; \boldsymbol{x}) \mathbf{1}_S(\boldsymbol{x}) d\boldsymbol{x} \geq \alpha/2 \,.$$

We begin with a claim about the shape of the worst case set. Let $t = (\|\boldsymbol{\mu}\|_2^2 - \|\boldsymbol{\mu}^\star\|_2^2)/2$ be the hyperplane threshold.

Claim 6. *Let $\mathcal{H}^+ = \{\boldsymbol{x} : \boldsymbol{x}^T(\boldsymbol{\mu} - \boldsymbol{\mu}^\star) < t\}$ and $\mathcal{H}^- = \{\boldsymbol{x} : \boldsymbol{x}^T(\boldsymbol{\mu} - \boldsymbol{\mu}^\star) > t\}$. The mass of the solution of the fractional knapsack is totally contained in either $\mathcal{H}^+$ or $\mathcal{H}^-$.*

Since the partition distribution satisfies Equation (4.4) with respect to the true Gaussian $\mathcal{N}(\boldsymbol{\mu}^\star)$ and since the set $\mathcal{H}$ is a hyperplane, the probability mass that is not cut by $\mathcal{H}$ is at least α. Hence, there exists a halfspace (either $\mathcal{H}^+$ or $\mathcal{H}^-$) with mass at least $\alpha/2$. Also, observe that the hyperplane $\mathcal{H}$ is the zero locus of the polynomial $q(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{\mu}\|_2^2 - \|\boldsymbol{x} - \boldsymbol{\mu}^\star\|_2^2$ and, hence, it is the set of points where the two spherical Gaussians $\mathcal{N}(\boldsymbol{\mu})$ and $\mathcal{N}(\boldsymbol{\mu}^\star)$ assign equal mass. We have that

$$\mathcal{H}^+ = \left\{\boldsymbol{x} : \mathcal{N}(\boldsymbol{\mu}^\star) > \mathcal{N}(\boldsymbol{\mu})\right\}.$$

Hence, we can assume that the worst-case set lies totally in $\mathcal{H}^+$ and, then, the optimization problem can be written as

$$\min_S \int \left(1 - \frac{\mathcal{N}(\boldsymbol{\mu}; \boldsymbol{x})}{\mathcal{N}(\boldsymbol{0}; \boldsymbol{x})}\right) \mathcal{N}(\boldsymbol{0}; \boldsymbol{x}) \mathbf{1}_S(\boldsymbol{x}) d\boldsymbol{x}\,,$$

$$\text{subj. to} \quad \int \mathcal{N}(\boldsymbol{0}; \boldsymbol{x}) \mathbf{1}_S(\boldsymbol{x}) d\boldsymbol{x} \geq \alpha/2, \quad S \in \mathcal{H}^+\,.$$

Without loss of generality, we assume that $\mathcal{N}^\star = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\mathcal{N} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I})$. In order to design the worst-case set, since the optimization has the structure of the fractional knapsack problem, we can think of each point $\boldsymbol{x} \in \mathcal{H}^+$ as having *weight* equal to its contribution to the mass gap $(\mathcal{N}(\boldsymbol{0}; \boldsymbol{x}) - \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{x}))$ and *value* equal to its density with respect to the true Gaussian $\mathcal{N}(\boldsymbol{0}; \boldsymbol{x})$. Hence, in order to design the worst-case set, the points $\boldsymbol{x} \in \mathcal{H}^+$ should be included in the set in order of increasing ratio of weight over value, until reaching a threshold T. So, we can define the worst-case set to be

$$S = \left\{\boldsymbol{x} \in \mathcal{H}^+ : 1 - \frac{\mathcal{N}(\boldsymbol{\mu}; \boldsymbol{x})}{\mathcal{N}(\boldsymbol{0}; \boldsymbol{x})} \leq T\right\} = \left\{\boldsymbol{x} \in \mathcal{H}^+ : 1 - \exp(p(\boldsymbol{x})) \leq T\right\},$$

where $p(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{x})^T(\boldsymbol{\mu} - \boldsymbol{x}) + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{x} = -\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} + \boldsymbol{\mu}^T\boldsymbol{x}$ and note that $p(\boldsymbol{x}) \leq 0$ for any $\boldsymbol{x} \in \mathcal{H}^+$. We will use the following anti-concentration result about the Gaussian mass of sets, defined by polynomials.

Lemma 4.2.14 (Theorem 8 of (CW01)). *Let $q, \gamma \in \mathbb{R}_+, \boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}$ in the positive semidefinite cone $\mathbb{S}_+^d$. Consider $p : \mathbb{R}^d \to \mathbb{R}$ a multivariate polynomial of degree at most ℓ and let*

$$\mathcal{Q} = \left\{\boldsymbol{x} \in \mathbb{R}^d : |p(\boldsymbol{x})| \leq \gamma\right\}.$$

Then, there exists an absolute constant C such that

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathcal{Q}) \leq \frac{Cq\gamma^{1/\ell}}{(\mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}[|p(\boldsymbol{z})|^{q/\ell}])^{1/q}}\,.$$

We can apply Lemma 4.2.14 for the quadratic polynomial $p(\boldsymbol{x})$ by setting $\gamma = \frac{\alpha^2}{256C^2}\sqrt{\mathbb{E}_{\boldsymbol{x} \sim \mathcal{N}^\star}[p^2(\boldsymbol{x})]}$ with $q = 4$, where C is the absolute Carbery-Wright constant. Hence, we get that the Gaussian mass of the set $\mathcal{Q} = \{\boldsymbol{x} : |p(\boldsymbol{x})| \leq \gamma\}$ is equal to

$$\mathcal{N}^\star(\mathcal{Q}) \leq \alpha/4\,.$$

So, for any point $\boldsymbol{x}$ in the remaining $\alpha/4$ mass of the set S, it holds that $|p(\boldsymbol{x})| \geq \gamma$. We first observe that γ can lower bounded by the total variation distance of $\mathcal{N}^\star$ and $\mathcal{N}$. It suffices to lower bound the expectation $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}^\star}[p^2(\boldsymbol{x})]$. We have that

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}^\star}\left[p^2(\boldsymbol{x})\right] \geq \mathop{\mathrm{Var}}_{\boldsymbol{x}\sim\mathcal{N}^\star}\left[p(x)\right] = \mathop{\mathrm{Var}}_{\boldsymbol{x}\sim\mathcal{N}^\star}\left[-\frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu} + \boldsymbol{\mu}^T\boldsymbol{x}\right] = \|\boldsymbol{\mu}\|_2^2\,,$$

and, hence

$$\gamma \geq \frac{\alpha^2}{256C^2}\cdot\|\boldsymbol{\mu}\|_2\,.$$

We will use the following lemma for the total variation distance of two Normal distributions.

Lemma 4.2.15 (see Corollaries 2.13 and 2.14 of (DKK+16))**.** *Let* $N_1 = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), N_2 = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ *be two Normal distributions. Then, it holds*

$$d_{\mathrm{TV}}(N_1, N_2) \leq \frac{1}{2}\left\|\boldsymbol{\Sigma}_1^{-1/2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right\|_2 + \sqrt{2}\left\|\boldsymbol{I} - \boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1/2}\right\|_F\,.$$

Applying Lemma 4.2.15 to the above inequality, we get

$$\gamma \geq \frac{\alpha^2}{256C^2}\cdot d_{\mathrm{TV}}(\mathcal{N}(\boldsymbol{\mu}), \mathcal{N}(\boldsymbol{\mu}^\star))\,.$$

To conclude, we have to lower bound the L_1 gap between $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}; \boldsymbol{x})\boldsymbol{1}_S(\boldsymbol{x})$ and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}; \boldsymbol{x})\boldsymbol{1}_S(\boldsymbol{x})$ and since S lies totally in $\mathcal{H}^+$

$$\int_S (\mathcal{N}(\boldsymbol{0};\boldsymbol{x}) - \mathcal{N}(\boldsymbol{\mu};\boldsymbol{x}))d\boldsymbol{x} = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}^\star}\left[1 - \exp(p(\boldsymbol{x}))\middle|\boldsymbol{1}_S(\boldsymbol{x})\right]\,.$$

To proceed, we distinguish two cases: First, assume that $\gamma \leq 1$ and recall that $\mathcal{Q} = \{\boldsymbol{x} : |p(\boldsymbol{x})| \leq \gamma\}$. Note that for $y \in [-1, 0]$, it holds that $1 - \exp(y) \geq |y|/2$ and, hence, we have that:

$$\int_S (\mathcal{N}(\boldsymbol{0};\boldsymbol{x}) - \mathcal{N}(\boldsymbol{\mu};\boldsymbol{x}))d\boldsymbol{x} \geq \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}^\star}\left[\frac{|p(\boldsymbol{x})|}{2}\boldsymbol{1}_{S\setminus\mathcal{Q}}(\boldsymbol{x})\right] \geq \gamma\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}^\star}\left[\boldsymbol{1}_{S\setminus\mathcal{Q}}(\boldsymbol{x})\right] \geq \frac{\alpha\gamma}{4}\,,$$

and, by the lower bound for γ, we get

$$\int_S (\mathcal{N}(\boldsymbol{0}, \boldsymbol{I};\boldsymbol{x}) - \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I};\boldsymbol{x}))d\boldsymbol{x} \geq C_\alpha \cdot d_{\mathrm{TV}}(\mathcal{N}(\boldsymbol{\mu}), \mathcal{N}(\boldsymbol{\mu}^\star))\,,$$

for some $C_\alpha = \Omega(\alpha^3)$. Otherwise, let $\gamma > 1$. Note that for $y < -1$, it holds that $1 - \exp(y) \geq 1/2$. Hence, we get that

$$\int_S (\mathcal{N}(\boldsymbol{0};\boldsymbol{x}) - \mathcal{N}(\boldsymbol{\mu};\boldsymbol{x}))d\boldsymbol{x} \geq \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}^\star}\left[\frac{1}{2}\boldsymbol{1}_{S\setminus\mathcal{Q}}(\boldsymbol{x})\right] \geq \alpha/8\,.$$

In conclusion, we get that

$$d_{\mathrm{TV}}(\mathcal{N}_\pi^*, \mathcal{N}_\pi) \geq C_\alpha \cdot d_{\mathrm{TV}}(\mathcal{N}^\star, \mathcal{N})\,,$$

where $C_\alpha = \mathrm{poly}(\alpha)$ and depends only on α. □

Chapter 5

Learning with Bounded Noise

5.1 Main Definitions and Results

We remind the reader some basic notions introduced in Section 2.4.3. Label Ranking is the problem of learning a hypothesis that maps features to rankings over a finite set of labels. Given a feature vector $\boldsymbol{x} \in \mathbb{R}^d$, a sorting function $\sigma(\cdot)$ maps it to a ranking of k alternatives, i.e., $\sigma(\boldsymbol{x})$ is an element of the symmetric group with k elements, $\mathbb{S}_k$. We focus on the fundamental concept class of Linear Sorting functions (HPRZ03). A linear sorting function parameterized by a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ with k rows $\boldsymbol{W}_1, \dots, \boldsymbol{W}_k$ takes a feature $\boldsymbol{x} \in \mathbb{R}^d$, maps it to $\boldsymbol{W}\boldsymbol{x} = (\boldsymbol{W}_1 \cdot \boldsymbol{x}, \dots, \boldsymbol{W}_k \cdot \boldsymbol{x}) \in \mathbb{R}^k$ and then outputs an ordering $(i_1, \dots, i_k)$ of the k alternatives such that $\boldsymbol{W}_{i_1} \cdot \boldsymbol{x} \geq \boldsymbol{W}_{i_2} \cdot \boldsymbol{x} \geq \dots \geq \boldsymbol{W}_{i_k} \cdot \boldsymbol{x}$.

We consider the natural setting where the feature vector $\boldsymbol{x} \in \mathbb{R}^d$ is generated by a standard normal distribution and the ground-truth ranking for each sample $\boldsymbol{x}$ is given by the LSF $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$ for some unknown parameter matrix $\boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$. For a fixed $\boldsymbol{x}$, the ranking that we observe comes from an η-noisy ranking distribution with ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$.

In particular, we observe samples generated as follows.

Definition 5.1.1 (Noisy Linear Label Ranking Distribution)**.** *Fix $\eta \in [0, 1/2)$ and some ground-truth parameter matrix $\boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$. We assume that the η-**noisy linear label ranking distribution** $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{S}_k$ satisfies the following:*

1. *The $\boldsymbol{x}$-marginal of $\mathcal{D}$ is the d-dimensional standard normal distribution.*
2. *For any $(\boldsymbol{x}, \pi) \sim \mathcal{D}$, the distribution of π conditional on $\boldsymbol{x}$ is an η-noisy ranking distribution with ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$, i.e., for any $i, j \in [k]$, with $i \neq j$, satisfies $\mathbf{Pr}_{\pi \sim \mathcal{M}(\sigma^\star)}[i \prec_\pi j \mid i \succ_{\sigma^\star} j] \leq \eta$.*

We refer to the above generative process as the bounded noise model (following the standard terminology in the literature). The main contributions of this chapter are the first efficient algorithms for learning LSFs with bounded noise with respect to Kendall's Tau distance and top-r disagreement loss.

Learning in Kendall's Tau Distance. The most standard metric in rankings (SSBD14) is Kendall's Tau (KT) distance which, for two rankings $\pi, \tau \in \mathbb{S}_k$, measures the fraction of pairs (i, j) on which they disagree. That is, $\Delta_{\mathrm{KT}}(\pi, \tau) = \sum_{i \prec_\pi j} \mathbf{1}\{i \succ_\tau j\} / \binom{k}{2}$. Our first result is an efficient learning algorithm that, given samples from an η-noisy linear label ranking distribution $\mathcal{D}$, computes a parameter matrix $\boldsymbol{W}$ that ranks the alternatives almost optimally with respect to the KT distance from the ground-truth ranking $\sigma_{\boldsymbol{W}^\star}(\cdot)$.

Theorem 5.1.2 (Learning LSFs in KT Distance)**.** *Fix $\eta \in [0, 1/2)$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1 with ground-truth LSF $\sigma_{\boldsymbol{W}^\star}(\cdot)$. There exists an algorithm that draws $N = \widetilde{O}\left(\frac{d}{\epsilon(1-2\eta)^6} \log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in sample-polynomial time, and computes a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ such that, with probability at least $1 - \delta$,*

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} [\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq \epsilon\,.$$

Our proper learning algorithm consists of two steps: an improper learning algorithm that decomposes the ranking problem to $O(k^2)$ binary linear classification problems and a convex (second order conic) program that "compresses" the k^2 linear classifiers to obtain a $k \times d$ matrix $\boldsymbol{W}$. Our improper learning algorithm splits the ranking learning problem into $O(k^2)$ binary, d-dimensional linear classification problems with Massart noise. In particular, for every pair of elements $i, j \in [k]$, each binary classification task asks whether element i is ranked higher than element j in the ground-truth permutation $\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$. As we already discussed, we have that, under the Gaussian distribution, there exist efficient Massart learning algorithms (BZ17; MV19; DKTZ20; ZSA20; ZL21) that can recover linear classifiers $\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x})$ that correctly order the pair i, j for all $\boldsymbol{x}$ apart from a region of $O(\epsilon)$-Gaussian mass. However, we still need to aggregate the results of the *approximate* binary classifiers in order to obtain a ranking of the k alternatives for each $\boldsymbol{x}$. We first show that we can design a "voting scheme" that combines the results of the binary classifiers using an efficient constant factor approximation algorithm for the Minimum Feedback Arc Set (MFAS) problem (ACN08). This gives us an efficient but improper algorithm for learning LSFs in Kendall's Tau distance. In order to obtain a proper learning algorithm, we further "compress" the $O(k^2)$ approximate linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ and obtain a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ with the property that the difference of every two rows $\boldsymbol{W}_i - \boldsymbol{W}_j$ is $O(\epsilon)$-close to the vector $\boldsymbol{v}_{ij}$. More precisely, we show that, given the linear classifiers $\boldsymbol{v}_{ij} \in \mathbb{R}^d$, we can efficiently compute a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ such that the following angle distance with $\boldsymbol{W}^\star$ is small:

$$d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star) \triangleq \max_{i,j} \theta(\boldsymbol{W}_i - \boldsymbol{W}_j, \boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \leq O(\epsilon)\,. \tag{5.1}$$

It is not hard to show that, as long as the above angle metric is at most $O(\epsilon)$, then (in expectation over the standard Gaussian) Kendall's Tau distance between the LSFs is also $O(\epsilon)$. A key technical difficulty that we face in this reduction is bounding the "condition number" of the convex (second order conic) program that finds the matrix $\boldsymbol{W}$ given the vectors $\boldsymbol{v}_{ij}$, see Claim 8. Finally, we remark that the proper learning algorithm of Theorem 5.1.2 results in a compact and efficient sorting function that requires: (i) storing $O(k)$ weight vectors as opposed to the initial $O(k^2)$ vectors of the improper learner; and (ii) evaluating k inner products with $\boldsymbol{x}$ to find its ranking (instead of $O(k^2)$).

Learning in top-r Disagreement. We next present our learning algorithm for the top-r metric formally defined as $\Delta_{\text{top}-r}(\pi, \tau) = \mathbf{1}\{\pi_{1..r} \neq \tau_{1..r}\}$, where by $\pi_{1..r}$ we denote the ordering on the first r elements of the permutation π. The top-r metric is a disagreement metric in the sense that it takes binary values and for $r = 1$ captures the standard (multiclass) top-1 classification loss. We remark that, in contrast with the top-r classification loss, which only requires the predicted label to be in the top-r predictions of the model, the top-r ranking metric that we consider here requires that the model puts *the same elements in the same order* as the ground truth in the top-r positions.

Theorem 5.1.3 (Learning LSFs in top-r Disagreement)**.** *Fix $\eta \in [0, 1/2)$, $r \in [k]$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1 with ground-truth LSF $\sigma_{\boldsymbol{W}^\star}(\cdot)$. There exists an algorithm that draws $N = \widetilde{O}\left(\frac{drk}{\epsilon(1-2\eta)^6}\log(1/\delta)\right)$ samples from $\mathcal{D}$, runs in sample-polynomial time and computes a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1 - \delta$,*

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\text{top}-r}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq \epsilon\,.$$

Suppose that we are interested in recovering only the top element of the ranking ($r = 1$). One approach would be to directly use the improper learning algorithm for this task and ask for KT distance of order roughly ϵ/k^2. The resulting hypothesis would produce good predictions for the top element but the required sample complexity would be $O(dk^2)$. While it seems that training $O(k^2)$ d-dimensional binary classifiers inherently requires $O(dk^2)$ samples, we show that, using the proper KT distance learning algorithm of Theorem 5.1.2, we can also obtain improved sample complexity results for the top-r metric. Our main technical contribution here is a novel estimate of the top-r disagreement in terms of the angle metric. In general, one can show that the top-r disagreement is at most $O(k^2)\ d_{\text{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)$. We significantly sharpen this estimate by showing the following lemma.

Lemma 5.1.4 (Top-r Disagreement via Parameter Distance)**.** *Consider two matrices $\boldsymbol{W}, \boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$ and let $\mathcal{N}_d$ be the standard Gaussian in d dimensions. We have that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{W}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{W}^\star\boldsymbol{x})] \leq \widetilde{O}(kr)\ d_{\text{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)\,.$$

We remark that Lemma 5.1.4 is a general geometric tool that we believe will be useful in other distribution-specific multiclass learning settings. The proof of Lemma 5.1.4 mainly relies on geometric Gaussian surface area computations that we believe are of independent interest. For the details, we refer the reader to Section 5.5. An interesting question with a convex-geometric flavor is whether the sharp bound of Lemma 5.1.4 also holds under the more general class of isotropic log-concave distributions.

5.2 Related Work

Robust Supervised Learning. We start with a summary of prior work on PAC learning with Massart noise. The Massart noise model was formally defined in (MN06) but similar variants had been defined by Vapnik, Sloan and Rivest (Vap06; Slo88; Slo92; RS94; Slo96). This model is a strict extension of the Random Classification Noise (RCN) model (Ang88), where the label noise is uniform, i.e., context-independent and is a special case of the agnostic model (Hau18; KSS94), where the label noise is fully adversarial and computational barriers are known to exist (GR09; FGKP06; Dan16; DKZ20; GGK20; DKPZ21; HSSVG22). Our work partially builds upon on the algorithmic task of PAC learning halfspaces with Massart noise (BH20). In the distribution-independent setting, known efficient algorithms (DGT19b; CKMY20; DKT21) achieve error $\eta+\epsilon$ and the works of (DK20; NT22) indicate that this error bound is the best possible in the Statistical Query model (Kea98). This lower bound motivates the study of the distribution-specific setting (which is also the case of our work). There is an extensive line of work in this direction: (ABHU15; ABHZ16; YZ17; ZLC17; BZ17; MV19; DKTZ20; ZSA20; ZL21) with the currently best algorithms succeeding for all $\eta < 1/2$ with a sample and computational complexity $\text{poly}(d, 1/\epsilon, 1/(1-2\eta))$ under a class of distributions including isotropic log-concave distributions. For details, see (DKK$^+$21). In this chapter we focus on Gaussian marginals but some of our results extend to larger distribution classes.

Label Ranking. Our results of this chapter lie in the area of Label Ranking, which has received significant attention over the years (SS07; HFCB08; CH08; HPRZ03; FHMB08; DSM03). There are multiple approaches for tackling this problem (see (VG10), (ZLY$^+$14)). Some of them are based on probabilistic models (CH08; CDH10; GDV12; ZLGQ14) or may be tree based, such as decision trees (CHH09), entropy based ranking trees and forests (RdSRSK15; dSSKC17), bagging techniques (AGM17) and random forests (ZQ18). There are also works focusing on supervised clustering (GDGV13). Finally, (CH08; CDH10; CHH09) adopt an instance-based approaches using nearest neighbors approaches. The above results are industrial. From a theoretical perspective, LR has been mainly studied from a statistical learning theory framework (CV20; CKS18; KGB18; KCS17). (FKP21) provide some computational guarantees for the performance of decision trees in the noiseless case and some experimental results on the robustness of random forests to noise. The setting of (DGR$^+$14) is close to ours but is investigated from an experimental standpoint. We remark that while reducing LR to multiple binary classification tasks has been used in prior literature (HFCB08; CH12; FKP21), standard reductions can not tolerate noise in rankings (nevertheless, from an experimental perspective, e.g., random forests seem robust to noise but lack formal theoretical guarantees). Our reduction crucially relies on the existence of efficient learning algorithms for binary linear classification with Massart noise.

5.3 Notation and Preliminaries

General Notation. We use $\widetilde{O}(\cdot)$ to omit poly-logarithmic factors. A learning algorithm has sample-polynomial runtime if it runs in time polynomial in the size of the description of the input training set. We denote vectors by boldface $\boldsymbol{x}$ (with elements x_i) and matrices with $\boldsymbol{W}$, where we let $\boldsymbol{W}_i \in \mathbb{R}^d$ denote the i-th row of $\boldsymbol{W} \in \mathbb{R}^{k \times d}$ and W_{ij} its elements. We denote $\boldsymbol{a} \cdot \boldsymbol{b}$ the inner product of two vectors and $\theta(\boldsymbol{a}, \boldsymbol{b})$ their angle. Let $\mathcal{N}_d$ denote the d-dimensional standard normal and $\Gamma(\cdot)$ the Gaussian surface area.

Rankings. We let $\mathrm{argsort}_{i \in [k]} \boldsymbol{v}$ denote the ranking of $[k]$ in decreasing order according to the values of $\boldsymbol{v}$. For a ranking π, we let $\pi(i)$ denote the position of the i-th element. If $\pi = \pi(\boldsymbol{x})$, we may also write $\pi(\boldsymbol{x})(i)$ to denote the position of i. We often refer to the elements of a ranking as *alternatives*. For a ranking σ, we let $\sigma_{1..r}$ denote the top-r part of σ. When $\sigma = \sigma(\boldsymbol{x})$, we may also write $\sigma_{1..r}(\boldsymbol{x})$ and $\sigma_\ell(\boldsymbol{x})$ will be the alternative at the ℓ-th position. We let Δ_{KT} denote the (normalized) KT distance, i.e., $\Delta_{\mathrm{KT}}(\pi, \tau) = \sum_{i \prec_\pi j} \mathbf{1}\{i \succ_\tau j\} / \binom{k}{2}$ for $\pi, \tau \in \mathbb{S}_k$.

5.4 Learning in KT distance: Theorem 5.1.2

In this section, we present the main tools required to obtain our proper learning algorithm of Theorem 5.1.2. Our proper algorithm adopts a two-step approach: it first invokes an efficient *improper* algorithm which, instead of a linear sorting function (i.e., a matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$), outputs a list of $O(k^2)$ linear classifiers. We then design a novel convex program in order to find the matrix $\boldsymbol{W}$ satisfying the guarantees of Theorem 5.1.2. Let us begin with the improper learner for LSFs with bounded noise with respect to the KT distance, whose description can be found in Algorithm 6.

5.4.1 Improper Learning Algorithm

Let us assume that the target function is $\sigma^\star(\boldsymbol{x}) = \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}) = \mathrm{argsort}(\boldsymbol{W}^\star \boldsymbol{x})$ for some $\boldsymbol{W}^\star \in \mathbb{R}^{k \times d}$.

Algorithm 6 Non-proper Learning Algorithm `ImproperLSF`

```
Input: Training set $T = \{(\boldsymbol{x}^t, \pi^t)\}_{t \in [N]}, \epsilon, \delta \in (0,1), \eta \in [0, 1/2)$
Output: Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$

For any $1 \leq i < j \leq k$, create $T_{ij} = \{(\boldsymbol{x}^t, \mathrm{sgn}(\pi^t(i) - \pi^t(j)))\}$
For any $1 \leq i < j \leq k$, compute $\boldsymbol{v}_{ij} = \mathtt{MassartLTF}(T_{ij}, \frac{\epsilon}{4}, \frac{\delta}{10k^2}, \eta)$    ▷ See Appendix 5.6.1
Ranking Phase: Given $\boldsymbol{x} \in \mathbb{R}^d$:
        (a) Construct directed graph $G$ with $V(G) = [k]$ and edges $e_{i \to j}$ only if $\boldsymbol{v}_{ij} \cdot \boldsymbol{x} > 0 \; \forall i \neq j$
        (b) Output $h(\boldsymbol{x}) = \mathtt{MFAS}(G)$    ▷ See Appendix 5.6.1
```

Step 1: Binary decomposition and Noise Structure. For each drawn example $(\boldsymbol{x}, \pi)$ from the η-noisy linear label ranking distribution $\mathcal{D}$ (see Definition 5.1.1), we create $\binom{k}{2}$ binary examples $(\boldsymbol{x}, y_{ij})$ with $y_{ij} = \mathrm{sgn}(\pi(i) - \pi(j))$ for any $1 \leq i < j \leq k$. We have that

$$\underset{(\boldsymbol{x},\pi)\sim\mathcal{D}}{\mathbf{Pr}}\left[y_{ij} \cdot \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x}) < 0 \mid \boldsymbol{x}\right] = \underset{\pi \sim \mathcal{M}(\sigma^\star(\boldsymbol{x}))}{\mathbf{Pr}}\left[\pi(i) < \pi(j) \mid \boldsymbol{W}_i^\star \cdot \boldsymbol{x} < \boldsymbol{W}_j^\star \cdot \boldsymbol{x}\right] .$$

Since $\mathcal{M}(\sigma^\star(\boldsymbol{x}))$ is an η-noisy ranking distribution (see Definition 2.4.3), we get that the above quantity is at most $\eta < 1/2$. Therefore, each sample $(\boldsymbol{x}, y_{ij})$ can be viewed as a sample from a distribution $\mathcal{D}_{ij}$ with Gaussian $\boldsymbol{x}$-marginal, optimal linear classifier $\mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})$, and Massart noise η. Hence, we have reduced the task of learning noisy LSFs to a number of $\binom{k}{2}$ sub-problems concerning the learnability of halfspaces in the presence of bounded (Massart) noise.

Step 2: Solving Binary Sub-problems. We can now apply the algorithm `MassartLTF` for LTFs with Massart noise under standard Gaussian marginals (ZSA20) (for details, see Appendix 5.6.1): for all the pairs of alternatives $1 \leq i < j \leq k$ with accuracy parameter ϵ', confidence $\delta' = O(\delta/k^2)$, and a total number of $N = \widetilde{\Omega}\left(\frac{d}{\epsilon'(1-2\eta)^6} \log(k/\delta)\right)$ i.i.d. samples from $\mathcal{D}$, we can obtain a collection of linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ for any $i < j$. We remark that each one of these halfspaces $\boldsymbol{v}_{ij}$ achieves ϵ disagreement with the ground-truth halfspaces $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ with high probability, i.e.,

$$\underset{\boldsymbol{x}\sim\mathcal{N}_d}{\mathbf{Pr}}[\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})] \leq \epsilon' .$$

Step 3: Ranking Phase. We now have to aggregate the linear classifiers and compute a single sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$. Given an example $\boldsymbol{x}$, we create the tournament graph G with k nodes that contains a directed edge $e_{i \to j}$ if $\boldsymbol{v}_{ij} \cdot \boldsymbol{x} > 0$. If G is acyclic, we output the induced permutation; otherwise, the graph contains cycles which should be eliminated. In order to output a ranking, we remove cycles from G with an efficient, 3-approximation algorithm for MFAS (ACN08; VZW09). Hence, the output $h(\boldsymbol{x})$ and the true target $\sigma^\star(\boldsymbol{x})$ will have $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(h(\boldsymbol{x}),\sigma^\star(\boldsymbol{x}))] \leq \epsilon' + 3\epsilon' = 4\epsilon'$. This last equation indicates why a constant factor approximation algorithm suffices for our purposes – we can always pick $\epsilon' = \epsilon/4$ and complete the proof. For details, see Appendix 5.6.1.

5.4.2 Proper Learning Algorithm: Theorem 5.1.2

Having obtained the improper learning algorithm, we can now describe our proper Algorithm 7. Initially, the algorithm starts similarly with the improper learner and obtains a collection of binary linear classifiers. The crucial idea is the next step: the design of an appropriate convex program which will efficiently give the matrix $\boldsymbol{W}$. We proceed with the details. For the proof, see Appendix 5.6.2.

Algorithm 7 Proper Learning Algorithm ProperLSF

```
Input: Training set $T = \{(\boldsymbol{x}^t, \pi^t)\}_{t\in[N]}, \epsilon, \delta \in (0,1), \eta \in [0,1/2)$
Output: Linear Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$, i.e., $h(\cdot) = \sigma_{\boldsymbol{W}}(\cdot)$ for some matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$

Compute $(\boldsymbol{v}_{ij})_{1\leq i<j\leq k}$ = ImproperLSF$(T, \epsilon, \delta, \eta)$        ▷ See Algorithm 6
Setup the CP 5.1 and compute $\boldsymbol{W}$ = Ellipsoid(CP)        ▷ See Appendix 5.6.2
Ranking Phase:    Given $\boldsymbol{x} \in \mathbb{R}^d$, output $h(\boldsymbol{x}) = \mathrm{argsort}(\boldsymbol{W}\boldsymbol{x})$
```

Step 1: Calling Non-proper Learners. As a first step, the algorithm calls Algorithm 6 with parameters ϵ, δ and $\eta \in [0,1/2)$ and obtains a list of linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ for $i<j$. Without loss of generality, assume that $\|\boldsymbol{v}_{ij}\|_2 = 1$.

Step 2: Designing and Solving the CP 5.1. Our main goal is to find a matrix $\boldsymbol{W}$ whose LSF is close to the true target in KT distance. We show the following lemma that connects the KT distance between two LSFs with the angle metric $d_{\mathrm{angle}}(\cdot,\cdot)$ defined in Eq. (5.1). The proof can be found in the Appendix 5.6.2.

Lemma 5.4.1. *For $\boldsymbol{W}, \boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$, it holds $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)$.*

The above lemma states that, for our purposes, it suffices to control the d_{angle} metric between the guess $\boldsymbol{W}$ and the true matrix $\boldsymbol{W}^\star$. It turns out that, given the binary classifiers $\boldsymbol{v}_{ij}$, we can design a convex program whose solution will satisfy this property. Thinking of the binary classifier $\boldsymbol{v}_{ij}$ as a proxy for $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$, we want each difference $\boldsymbol{W}_i - \boldsymbol{W}_j$ to have small angle with $\boldsymbol{v}_{ij}$ or equivalently to have large correlation with it, i.e., $(\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} \approx \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2$. To enforce this condition, we can therefore use the second order conic constraint $(\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} \geq (1-\phi)\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2$. We formulate the following convex program 5.1 with variable the matrix $\boldsymbol{W}$:

$$\begin{aligned} &\text{Find} && \boldsymbol{W} \in \mathbb{R}^{k\times d}, \quad \|\boldsymbol{W}\|_F \leq 1, \\ &\text{such that} && (\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} \geq (1-\phi) \cdot \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 \quad \text{for any } 1 \leq i < j \leq k, \end{aligned} \tag{5.1}$$

for some $\phi \in (0,1)$ to be decided. Intuitively, since any $\boldsymbol{v}_{ij}$ has good correlation with $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ (by the guarantees of the improper learning algorithm) and the CP 5.1 requires that its solution $\boldsymbol{W}$ similarly correlates well with $\boldsymbol{v}_{ij}$, we expect that $d_{\text{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)$ will be small. We show that:

Claim 7. *The convex program 5.1 is feasible and any solution $\boldsymbol{W}$ of 5.1 satisfies $d_{\text{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star) \leq \epsilon$.*

To see this, note that any solution of CP 5.1 is a matrix $\boldsymbol{W}$ whose angle metric (see Eq. (5.1)) with the true matrix is small by an application of the triangle inequality between the angles of $(\boldsymbol{v}_{ij}, \boldsymbol{W}_i - \boldsymbol{W}_j)$ and $(\boldsymbol{v}_{ij}, \boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)$ for any $i \neq j$. We next have to deal with the feasibility of CP 5.1. Our goal is to determine the value of ϕ that makes the CP 5.1 feasible. For the pair $1 \leq i < j \leq k$, the guess $\boldsymbol{v}_{ij}$ and the true normal vector $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ satisfy, with high probability,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{D}_x}[\operatorname{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \operatorname{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})] \leq \epsilon\,. \tag{5.2}$$

Under the Gaussian distribution (which is rotationally symmetric), it is well known that the angle $\theta(\boldsymbol{u}, \boldsymbol{v})$ between two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ is equal to $\pi \cdot \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[\operatorname{sgn}(\boldsymbol{u} \cdot \boldsymbol{x}) \neq \operatorname{sgn}(\boldsymbol{v} \cdot \boldsymbol{x})]$. Hence, using Eq. (5.2), we get that the angle between the guess $\boldsymbol{v}_{ij}$ and the true normal vector $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ is $\theta(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star, \boldsymbol{v}_{ij}) \leq c\epsilon$. For sufficiently small ϵ, this bound implies that the cosine of the above angle is of order $1-(c\epsilon)^2$ and so the following inequality will hold (since $\boldsymbol{v}_{ij}$ is unit):

$$(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{v}_{ij} \geq (1 - 2(c\epsilon)^2) \cdot \|\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star\|_2\,.$$

Hence, by setting $\phi = 2(c\epsilon)^2$, the convex program 5.1 with variables $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ will be feasible; since $\|\boldsymbol{W}^\star\|_F \leq 1$ comes without loss of generality, $\boldsymbol{W}^\star$ will be a solution with probability $1-\delta$.

Next, we have to control the volume of the feasible region. This is crucial in order to apply the ellipsoid algorithm (for details, see in Appendix 5.6.2) and, hence, solve the convex program. We show the following claim (see Appendix 5.6.2 for the proof):

Claim 8. *There exists $\rho \geq 2^{-\operatorname{poly}(d,k,1/\epsilon,\log(1/\delta))}$ so that the feasible set of CP 5.1 with $\phi = O(\epsilon^2)$ contains a ball (with respect to the Frobenius norm) of radius ρ.*

Critically, the runtime of the ellipsoid algorithm is *logarithmic* in $1/\rho$. So, the ellipsoid runs in time polynomial in the parameters of the problem and outputs the desired matrix $\boldsymbol{W}$.

5.5 Learning in top-r Disagreement: Theorem 5.1.3

In this section we show that the proper learning algorithm of Section 5.4.2 learns noisy LSFs in the top-r disagreement metric. We have seen that, with $\widetilde{O}(d \log(k)/\epsilon)$ samples, Algorithm 7 of Section 5.4.2 computes a matrix $\boldsymbol{W}$ such that $d_{\text{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star) \leq \epsilon$, see Claim 7. Let us be more specific. Lemma 5.4.1 relates the expected KT distance with the angle metric of the two matrices (see also Equation (5.1)). Our Algorithm 7 essentially gives an upper bound on this angle metric. When we shift our objective and our goal is to control the top-r disagreement, we can still apply Algorithm 7 which essentially controls the angle metric. The crucial ingredient that is missing is the relation between the loss we have to control, i.e., the expected top-r disagreement and the angle metric of Equation 5.1. This relation is presented right after and essentially says that the expected top-r disagreement is at most $O(kr)$ times this angle metric. Hence, in order to get top-r disagreement of order ϵ, it suffices to apply our Algorithm 7 with $\epsilon' = O(\epsilon/(kr))$.

We continue with our main contribution which is the following lemma that connects the top-r disagreement metric with the geometric distance $d_{\text{angle}}(\cdot, \cdot)$, recall Lemma 5.1.4. To keep this sketch simple we shall present a sketch of the proof of Lemma 5.1.4 for the special case of top-1 classification, which we restate below. The proof of the top-1 case can be found at the Appendix 5.7. The detailed proof of the general case ($r > 1$) can be found in the Appendix 5.8.

Lemma 5.5.1 (Top-1 Disagreement Loss via $d_{\text{angle}}(\cdot, \cdot)$)**.** *Consider two matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{k \times d}$ and let $\mathcal{N}_d$ be the standard Gaussian in d dimensions. We have that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] \leq O\left(k\sqrt{\log k}\right) \; d_{\text{angle}}(\boldsymbol{U}, \boldsymbol{V})\,.$$

We observe that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] = \sum_{i \in [k]} \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) = i, \sigma_1(\boldsymbol{V}\boldsymbol{x}) \neq i]\,. \tag{5.1}$$

We denote by $\mathcal{C}_{\boldsymbol{U}}^{(i)} \triangleq \mathbf{1}\{\boldsymbol{x} : \sigma_1(\boldsymbol{U}\boldsymbol{x}) = i\} = \prod_{j \neq i} \mathbf{1}\{(\boldsymbol{U}_i - \boldsymbol{U}_j) \cdot \boldsymbol{x} \geq 0\}$, i.e., this is the set where the ranking corresponding to $\boldsymbol{U}$ picks i as the top element. Note that $\mathcal{C}_{\boldsymbol{U}}^{(i)}$ is the indicator of a homogeneous polyhedral cone since it can be written as the intersection of homogeneous halfspaces. Using these cones we can rewrite the top-1 disagreement of Eq. (5.1) as

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] = \sum_{i \in [k]} \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[C_{\boldsymbol{U}}^{(i)}(\boldsymbol{x}) = 1, C_{\boldsymbol{V}}^{(i)}(\boldsymbol{x}) = 0]\,. \tag{5.2}$$

Hence, our task is to control the mass of the disagreement region of two cones. The next Lemma 5.5.2 achieves this task and, combined with Eq. (5.2) directly gives the conclusion of Lemma 5.5.1.

Next we work with two general homogeneous polyhedral cones with set indicators C_1, C_2:

Lemma 5.5.2 (Cone Disagreement)**.** *Let $C_1, C_2 : \mathbb{R}^d \mapsto \{0,1\}$ be homogeneous polyhedral cones defined by the k unit vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$ and $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ respectively. For some universal constant $c > 0$, it holds that $\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq c\sqrt{\log k}\ \max_{i \in [k]} \theta(\boldsymbol{v}_i, \boldsymbol{u}_i)$.*

Roadmap of the Proof of Lemma 5.5.2: Assume that we rotate one face of the polyhedral cone C_1 by a very small angle θ to obtain the perturbed cone C_2. At a high-level, we expect the probability of the disagreement region between the new cone C_2 and C_1 to be roughly (this is an underestimation) equal to the size of the perturbation θ times the (Gaussian) surface area of the face of the convex cone that we perturbed. The Gaussian Surface Area (GSA) of a convex set $A \subset \mathbb{R}^d$, is defined as $\Gamma(A) \triangleq \int_{\partial A} \phi_d(\boldsymbol{x}) d\mu(\boldsymbol{x})$, where $d\mu(\boldsymbol{x})$ is the standard surface measure in $\mathbb{R}^d$ and $\phi_d(\boldsymbol{x}) = (2\pi)^{-d/2} \cdot \exp(-\|\boldsymbol{x}\|_2^2/2)$. In fact, in Claim 9 below, we show that the probability of the disagreement between C_1 and C_2 is roughly $O(\theta)\Gamma(F_1)\sqrt{\log(1/\Gamma(F_1) + 1)}$, where F_1 is the face of cone C_1 that we rotated. Now, when we perturb all the faces by small angles (all perturbations are at most θ), we can show (via a sequence of triangle inequalities) that the total probability of the disagreement region is bounded above by the perturbation size θ times the sum of the Gaussian surface area of every face (times a logarithmic blow-up factor):

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq O(\theta) \sum_{i=1}^{k} \Gamma(F_i)\sqrt{\log(1/\Gamma(F_i) + 1)}\,.$$

Surprisingly, for homogeneous convex cones, the above sum cannot grow very fast with k. In fact, we show that it can be at most $O(\sqrt{\log k})$. To prove this, we crucially rely on the following convex geometry result showing that the Gaussian surface area of a homogeneous convex cone is $O(1)$ regardless of the number of its faces k.

Lemma 5.5.3 ((Naz03))**.** *Let C be a homogeneous polyhedral cone with k faces $F_1, \dots, F_k$. Then C has Gaussian surface area $\Gamma(C) = \sum_{i=1}^{k} \Gamma(F_i) \leq 1$.*

Using an inequality similar to the fact that the maximum entropy of a discrete distribution on k elements is at most $\log k$, and, since, from Lemma 5.5.3, it holds that $\sum_{i=1}^{k} \Gamma(F_i) \leq 1$, we can show that $\sum_{i=1}^{k} \Gamma(F_i)\sqrt{\log(1/\Gamma(F_i) + 1)} = O(\sqrt{\log k})$. Therefore, with the above lemma we conclude that, if the maximum angle perturbation that we perform on C_1 is θ, then the probability of the disagreement region is $O(\theta)$. We next give the formal proof resulting in the upper bound of $O(\sqrt{\log k}\ \theta)$ for the disagreement.

Single Face Perturbation Bound: Claim 9: We will use the following notation for the positive orthant indicator $R(\boldsymbol{z}) = \prod_{i=1}^{k} \mathbf{1}\{\boldsymbol{z}_i \geq 0\}$. Notice that the homogeneous polyhedral cone C_1 can be written as $C_1(\boldsymbol{x}) = R(\boldsymbol{V}\boldsymbol{x}) = R(\boldsymbol{v}_1 \cdot$

$\boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$. Claim 9 below shows that the disagreement of two cones that differ on a single normal vector is bounded by above by the Gaussian surface area of a particular face F_1 times a logarithmic blow-up factor $\sqrt{\log(1/\Gamma(F_1)+1)}$.

Claim 9. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{R}^d$ and $\boldsymbol{r} \in \mathbb{R}^d$ with $\theta(\boldsymbol{v}_1, \boldsymbol{r}) \leq \theta$ for some sufficiently small $\theta \in (0, \pi/2)$. Let F_1 be the face with $\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0$ of the cone $R(\boldsymbol{V}\boldsymbol{x})$ and $c > 0$ be some universal constant. Then,*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d} \left[R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq R(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\right] \leq c \cdot \theta \cdot \Gamma(F_1) \sqrt{\log\left(\frac{1}{\Gamma(F_1)} + 1\right)} .$$

Proof Sketch of Claim 9. Since the constraints $\boldsymbol{v}_2 \cdot \boldsymbol{x} \geq 0, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x} \geq 0$ are common in the two cones, we have that $R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq R(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$ only when the first "halfspaces" disagree, i.e., when $(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0$. Thus, we have that the LHS probability of Claim 9 is equal to

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} \left[R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \cdot \mathbf{1}\{(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0\}\right] . \tag{5.3}$$

This expectation contains two terms: the term $R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots \boldsymbol{v}_k \cdot \boldsymbol{x})$ that contains the last $k-1$ common constrains of the two cones and the region where the first two halfspaces disagree, i.e., the set $\{\boldsymbol{x} : (\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0\}$. In order to upper bound this integral in terms of the angle θ, we observe that (for θ sufficiently small) it is not hard to show (see Appendix B) that the disagreement region, which is itself a (non-convex) cone, is a subset of the region $\{\boldsymbol{x} : |\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq 2\theta |\boldsymbol{q} \cdot \boldsymbol{x}|\}$, where $\boldsymbol{q}$ the normalized projection of $\boldsymbol{r}$ onto the orthogonal complement of $\boldsymbol{v}_1$, i.e., $\boldsymbol{q} = \mathrm{proj}_{\boldsymbol{v}_1^\perp} \boldsymbol{r} / \|\mathrm{proj}_{\boldsymbol{v}_1^\perp} \boldsymbol{r}\|_2$. Therefore, we have that the integral of Eq. (5.3) is at most

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} \left[R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \ \mathbf{1}\{|\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq 2\theta |\boldsymbol{q} \cdot \boldsymbol{x}|\}\right] .$$

This is where the definition of the Gaussian surface area appears. In fact, we have to compute the derivative of the above expression (which is a function of θ) with respect to θ and evaluate it at $\theta = 0$. The idea behind this computation is that we can upper bound probability mass of the cone disagreement, i.e., the term $\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{N}_d} \left[R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq R(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\right]$ by its derivative with respect to θ (evaluated at 0) times θ by introducing $o(\theta)$ error. Hence, it suffices to upper bound the value of this derivative at 0, which is:

$$2 \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} \left[R(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \ |\boldsymbol{q} \cdot \boldsymbol{x}| \ \delta(|\boldsymbol{v}_1 \cdot \boldsymbol{x}|)\right] ,$$

where δ is the Dirac delta function. Notice that, if we did not have the term $|\boldsymbol{q} \cdot \boldsymbol{x}|$, the above expression would be exactly equal to two times the Gaussian surface area of the face with $\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0$, i.e., it would be equal to $2\Gamma(F_1)$. We now show that this extra term of $|\boldsymbol{q} \cdot \boldsymbol{x}|$ can only increase the above surface integral by at most a logarithmic factor. For some ξ to be decided, we have that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ |\boldsymbol{q}\cdot\boldsymbol{x}|\ \delta(|\boldsymbol{v}_1\cdot\boldsymbol{x}|)\right] = \int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|d\mu(\boldsymbol{x})\\
&\le \int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\le\xi\}d\mu(\boldsymbol{x}) + \int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\ge\xi\}d\mu(\boldsymbol{x})\\
&\le \xi\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})d\mu(\boldsymbol{x}) + \int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\ge\xi\}d\mu(\boldsymbol{x})\,,
\end{aligned}$$

where $d\mu(\boldsymbol{x})$ is the standard surface measure in $\mathbb{R}^d$. The first integral above is exactly equal to the Gaussian surface area of the face F_1. To bound from above the second term we can use the next claim showing that not a lot of mass of the face F_1 can concentrate on the region where $|\boldsymbol{q}\cdot\boldsymbol{x}|$ is very large. Its proof relies on standard Gaussian concentration arguments, and is provided in Appendix 5.7.

Claim 10. *It holds that* $\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\ge\xi\}d\mu(\boldsymbol{x})\le O(\exp(-\xi^2/2))\,.$

Using the above result, we get that

$$\frac{d}{d\theta}\left(\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ \mathbf{1}\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\le 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}\right]\right)\Big|_{\theta=0}\le O(\xi)\,\Gamma(F_1)+O(\exp(-\xi^2/2))\,.$$

By picking $\xi=\Theta(\sqrt{\log(1+1/\Gamma(F_1))})$, the result follows since, up to introducing $o(\theta)$ error, we can bound the term $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\neq R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\right]$ by its derivative with respect to θ, evaluated at 0, times θ. □

5.6 Learning LSFs with Bounded Noise in Kendall's Tau distance

5.6.1 Improperly Learning LSFs with Bounded Noise

We provide an improper learner for LSFs in the presence of bounded noise. We first restate the main result of this section, whose proof relies on a connection between noisy linear label ranking distributions and the Massart noise model.

Theorem 5.6.1 (Non-Proper Learning Algorithm)**.** *Fix $\eta\in[0,1/2)$ and $\epsilon,\delta\in(0,1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1.* `ImproperLSF` *(Algorithm 6) draws $N=\widetilde{O}\left(\frac{d}{\epsilon(1-2\eta)^6}\ \log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in* $\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))$ *time and, with probability at least $1-\delta$, outputs a hypothesis $h:\mathbb{R}^d\to\mathbb{S}_k$ that is ϵ-close in KT distance to the target.*

Proof. Assume that the target function is $\sigma^\star(\boldsymbol{x})=\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})=\mathrm{argsort}(\boldsymbol{W}^\star\boldsymbol{x})$ for some unknown matrix $\boldsymbol{W}^\star\in\mathbb{R}^{k\times d}$. Consider a collection of N i.i.d. samples from an η-noisy linear label ranking distribution $\mathcal{D}$ (see Definition 5.1.1) and let T be the associated training set. For each example $(\boldsymbol{x}, \pi) \in T$, we create a list of $\binom{k}{2}$ binary examples $(\boldsymbol{x}, y_{ij})$ with $y_{ij} = \mathrm{sgn}(\pi(i) - \pi(j))$ for any $1 \leq i < j \leq k$, where $\pi(i)$ denotes the position of the element i. Hence, we create the datasets T_{ij} consisting of the binary labeled examples $(\boldsymbol{x}, y_{ij})$. We have that

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\pi)\sim\mathcal{D}} \left[y_{ij} \cdot \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x}) < 0 \mid \boldsymbol{x}\right] = \mathop{\mathbf{Pr}}_{\pi\sim\mathcal{M}(\sigma^\star(\boldsymbol{x}))} \left[\pi(i) < \pi(j) \mid \boldsymbol{W}_i^\star \cdot \boldsymbol{x} < \boldsymbol{W}_j^\star \cdot \boldsymbol{x}\right] .$$

Since $\mathcal{M}(\sigma^\star(\boldsymbol{x}))$ is an η-bounded noise ranking distribution (see Definition 2.4.3), we get that

$$\mathop{\mathbf{Pr}}_{\pi\sim\mathcal{M}(\sigma^\star(\boldsymbol{x}))} [\pi(i) < \pi(j) \mid \sigma^\star(\boldsymbol{x})(i) > \sigma^\star(\boldsymbol{x})(j)] \leq \eta < 1/2 ,$$

where $\sigma^\star(\boldsymbol{x})(i)$ denotes the position of the element i in the ranking $\sigma^\star(\boldsymbol{x})$. Focusing on the training set T_{ij}, we have that the sign y_{ij} is flipped with probability at most η. So, we have reduced the problem to $\binom{k}{2}$ sub-problems concerning the learnability of halfspaces in the presence of Massart noise. The Massart noise model is a special case of Definition 5.1.1 where $k = 2$. Note also that for each training set T_{ij}, the features $\boldsymbol{x}$ have the same distribution. We can now apply the following result for LTFs with Massart noise for the standard Gaussian distribution. Recall that the concept class of homogeneous halfspaces (or linear threshold functions) is $\mathcal{C}_{\mathrm{LTF}} = \{h_{\boldsymbol{w}}(\boldsymbol{x}) = \mathrm{sgn}(\boldsymbol{w} \cdot \boldsymbol{x}) : \boldsymbol{w} \in \mathbb{R}^d\}$.

Lemma 5.6.2 (Learning Halfspaces with Massart noise (ZSA20))**.** *Fix $\eta \in [0, 1/2)$ and let $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1 with $k = 2$ (where $\mathcal{C}_{\mathrm{LSF}} = \mathcal{C}_{\mathrm{LTF}}$). There is a computationally efficient algorithm `MassartLTF` that draws $m = O(\frac{d \text{ polylog}(d)}{\epsilon(1-2\eta)^6} \cdot \log(1/\delta))$ samples from $\mathcal{D}$, runs in $\mathrm{poly}(m)$ time and outputs a linear threshold function h that is ϵ-close to the target linear threshold function $h^\star$ with probability at least $1 - \delta$, i.e., it holds $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[h(\boldsymbol{x}) \neq h^\star(\boldsymbol{x})] \leq \epsilon$.*

We can invoke the algorithm of Lemma 5.6.2 for any alternatives $1 \leq i < j \leq k$ with accuracy $\epsilon' = O(\epsilon)$, $\delta' = O(\delta/k^2)$ and error rate $\eta < 1/2$[1]. We remark that Lemma 5.6.2 returns a halfspace. Each one of the $\binom{k}{2}$ calls will provide a vector $\boldsymbol{v}_{ij} \in \mathbb{R}^d$ such that, with probability at least $1 - \delta'$, it satisfies

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d} [\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})] \leq \epsilon' ,$$

where the true target halfspace has normal vector $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$. Moreover, for any $i < j$, the algorithm requires that the training set T_{ij} is of size

$$|T_{ij}| = \Omega\left(\frac{d}{\epsilon'} \cdot \frac{1}{(1-2\eta)^6} \cdot \log(1/\delta')\right) ,$$

[1] We can assume that η is known without loss of generality.

and, so, a total number of

$$N = \Omega \left(\frac{d}{\epsilon} \cdot \frac{1}{(1-2\eta)^6} \cdot \log(k/\delta) \right) ,$$

samples $(\boldsymbol{x}, \pi)$ is required from the distribution $\mathcal{D}$. Given a collection of linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ for any $i < j$, it remains to aggregate them and compute a sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$. To this end, the estimator h, given an example $\boldsymbol{x}$, creates the directed complete graph G with k nodes with directed edge $i \to j$ if $\boldsymbol{v}_{ij} \cdot \boldsymbol{x} > 0$. If all the linear classifiers are correct (which occurs with probability $1 - O(\epsilon k^2)$ over $\mathcal{D}_x$ due to the union bound), the graph G is acyclic (since it will match the true directions induced by $\boldsymbol{W}^\star$) and the estimator h outputs the induced permutation. Observe that the KT distance is

$$\frac{1}{\binom{k}{2}} \cdot \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} \left[\sum_{1 \leq i < j \leq k} \mathbf{1}\{\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})\} \right] \leq \epsilon' .$$

Otherwise, the classifiers are inconsistent and G contains cycles. So, the expected number of mistakes in the graph G is ϵk^2. The estimator in order to output a ranking uses a deterministic constant approximation algorithm for the minimum Feedback Arc Set (ACN08) in order to remove the cycles. For an overview of this fundamental line of research, we refer to (ACN08; VZW09; KMS06).

Lemma 5.6.3 (3-Approximation Algorithm for mimimum FAS (see (VZW09; ACN08))). *There is a deterministic algorithm* `MFAS` *for the minimum Feedback Arc Set on unweighted tournaments with k vertices that outputs orderings with cost less than* $3 \cdot \mathrm{OPT}$. *The running time is* $\mathrm{poly}(k)$.

In the above, OPT is the minimum number of flips the algorithm should perform. With input the cyclic directed graph G induced by the estimated linear classifiers, the algorithm of Lemma 5.6.3 computes, in $\mathrm{poly}(k)$ time, a 3-approximation of the optimal solution (i.e., instead of correcting ϵ_0 directed edges, the algorithm will provide a directed acyclic graph with $3\epsilon_0$ changed edges). Hence, for the hypothesis $h : \mathbb{R}^d \to \mathbb{S}_k$, where $h(\boldsymbol{x})$ is the output of the minimum FAS approximation algorithm with input G (G depends on the input $\boldsymbol{x}$, the randomness of the samples and the internal randomness of the $\binom{k}{2}$ calls of the Massart linear classifiers), and the target function $\sigma^\star(\boldsymbol{x})$, we have that

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d} [\Delta_{KT}(h(\boldsymbol{x}), \sigma^\star(\boldsymbol{x}))] \leq (\epsilon' + 3\epsilon') = 4\epsilon' ,$$

which completes the proof, by setting $\epsilon' = \epsilon/4$. □

Remark 4. *Consider the following variant of the above procedure: compute the $O(k^2)$ linear classifiers with accuracy $\epsilon' = \epsilon/k^2$: If the induced directed graph is acyclic, output the ranking; otherwise, output a random permutation. With probability ϵ, the KT distance will be of order k^2. Hence, one has to draw in total $O(k^4 d/\epsilon)$ samples to make the expected KT distance roughly $O(\epsilon)$. The algorithm of Theorem 5.6.1 improves on this approach.*

5.6.2 The Proof of Theorem 5.1.2: Properly Learning LSFs with Bounded Noise

We first restate the main result of this section.

Theorem 5.6.4 (Proper Learning Algorithm). *Fix $\eta \in [0, 1/2)$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1. `ProperLSF` (Algorithm 7) draws $N = \widetilde{O}\left(\frac{d}{\epsilon(1-2\eta)^6} \log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$ time and, with probability at least $1 - \delta$, outputs a Linear Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$ that is ϵ-close in KT distance to the target.*

We are now ready to provide the proof of our efficient proper learning algorithm for the class of Linear Sorting functions in the presence of bounded noise with respect to the standard Gaussian probability measure.

Proof. As a first step, the algorithm calls the improper learning algorithm `ImproperLSF` (Algorithm 6) with parameters ϵ, δ and $\eta < 1/2$ and obtains a list of linear classifiers with normal vectors $\boldsymbol{v}_{ij}$ for $i < j$. The utility of this step implies that, with probability at least $1 - \delta$, each one of the classifiers ϵ-learns the associated true halfspace, i.e., it holds

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})] \leq \epsilon\,,$$

where $\boldsymbol{W}^\star$ is the matrix of the target Linear Sorting function. Without loss of generality, assume that $\|\boldsymbol{v}_{ij}\|_2 = 1$. In order to make the learner proper, it suffices to solve the following convex program on $\boldsymbol{W}$:

$$\begin{aligned} &\text{Find} && \boldsymbol{W} \in \mathbb{R}^{k \times d}, && (5.1)\\ &\text{such that} && (\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} \geq (1 - \phi) \cdot \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 \quad \text{for any } 1 \leq i < j \leq k\,, && (5.2)\\ &&& \|\boldsymbol{W}\|_F \leq 1\,, && (5.3) \end{aligned} \qquad (\mathtt{CP})$$

for some $\phi \in (0, 1)$ to be decided. The main key ideas are summarized in the next claim.

Claim 11. *The following properties hold true for $\phi = O(\epsilon^2)$ with probability at least $1 - \delta$.*

1. *The convex program 5.1 is feasible.*

2. *Any solution of the convex program 5.1 induces an LSF that is ϵ-close in KT distance to the true target $\sigma_{\boldsymbol{W}^\star}(\cdot)$.*

3. *The feasible set of the convex program 5.1 contains a ball of radius $r = 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ and is contained in a ball of radius 1. Both balls are with respect to the Frobenius norm.*

4. *The convex program 5.1 can be solved in time* $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$ *using the ellipsoid algorithm.*

Proof of Item 1. First, we can choose the error ϕ so that this convex program is feasible. Let us set $\boldsymbol{W} = \boldsymbol{W}^\star$, where $\boldsymbol{W}^\star$ is the underlying matrix of the target Linear Sorting function $\sigma^\star$ with $\sigma^\star(\boldsymbol{x}) = \mathrm{argsort}(\boldsymbol{W}^\star \boldsymbol{x})$. Recall that, by the guarantees of the improper learning algorithm, for the pair $1 \leq i < j \leq k$, it holds

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{v}_{ij} \cdot \boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{x})] \leq \epsilon\,. \tag{5.4}$$

Since the standard Gaussian is rotationally symmetric, the angle $\theta(\boldsymbol{u}, \boldsymbol{v})$ between two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ is equal to $\pi \cdot \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{u} \cdot \boldsymbol{x}) \neq \mathrm{sgn}(\boldsymbol{v} \cdot \boldsymbol{x})]$. Hence, using this observation and Equation (5.4), we get that the angle between the guess vector $\boldsymbol{v}_{ij}$ and the true normal vector $\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star$ is

$$\theta(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star, \boldsymbol{v}_{ij}) \leq c \cdot \epsilon\,,$$

for some constant $c > 0$. For sufficiently small ϵ, this bound implies that the cosine of the above angle is of order $1 - (c\epsilon)^2$ and so the following inequality will hold

$$(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star) \cdot \boldsymbol{v}_{ij} \geq (1 - 2(c\epsilon)^2) \cdot \|\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star\|_2\,,$$

since $\boldsymbol{v}_{ij}$ is unit. Hence, by setting $\phi = 2(c\epsilon)^2$, the convex program with variables $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ will be feasible; $\boldsymbol{W}^\star$ will be a solution with probability $1 - \delta$, where the randomness is over the output of the algorithm dealing with the Massart linear classifiers. Note that we can assume that $\|\boldsymbol{W}^\star\|_F \leq 1$ without loss of generality, since we can divide each row with the Frobenius norm.

Proof of Item 2. Let $\widetilde{\boldsymbol{W}}$ be a solution of the convex program.We will make use of the observation that the angle between two vectors is equal to the disagreement of the associated linear threshold functions with respect to the standard normal times π. Observe that any solution $\widetilde{\boldsymbol{W}}$ to the convex program will satisfy that

$$(\forall i, j) \quad \theta(\boldsymbol{v}_{ij}, \widetilde{\boldsymbol{W}}_i - \widetilde{\boldsymbol{W}}_j) \leq O(\sqrt{\phi}) = c\epsilon\,.$$

and

$$(\forall i, j) \quad \theta(\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star, \boldsymbol{v}_{ij}) \leq \epsilon\,.$$

This implies that

$$d_{\mathrm{angle}}(\boldsymbol{W}^\star, \widetilde{\boldsymbol{W}}) \leq c'\ \epsilon$$

Claim 12. *For the matrices* $\boldsymbol{W}, \boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$*, it holds that*

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}), \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] \leq d_{\mathrm{angle}}(\boldsymbol{W}, \boldsymbol{W}^\star)\,.$$

Proof. We have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\Delta_{\mathrm{KT}}(\sigma_{\boldsymbol{W}}(\boldsymbol{x}),\sigma_{\boldsymbol{W}^\star}(\boldsymbol{x}))] &= \frac{1}{\binom{k}{2}}\cdot \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sum_{1\le i<j\le k}\mathbf{1}\{((\boldsymbol{W}_i-\boldsymbol{W}_j)\cdot\boldsymbol{x})\ ((\boldsymbol{W}_i^\star-\boldsymbol{W}_j^\star)\cdot\boldsymbol{x})<0\}\\
&= \frac{1}{\binom{k}{2}}\cdot\sum_{1\le i<j\le k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{W}_i-\boldsymbol{W}_j)\cdot\boldsymbol{x})\neq\mathrm{sgn}((\boldsymbol{W}_i^\star-\boldsymbol{W}_j^\star)\cdot\boldsymbol{x})]\\
&= \frac{1}{\pi}\max_{i,j}\theta(\boldsymbol{W}_i-\boldsymbol{W}_j,\boldsymbol{W}_i^\star-\boldsymbol{W}_j^\star)\\
&\le d_{\mathrm{angle}}(\boldsymbol{W},\boldsymbol{W}^\star)\,.
\end{aligned}$$

□

Using the above claim, we get an expected KT distance bound of order $O(\epsilon)$. This gives the desired result.

Proof of Item 3. We will make use of the next lemma.

Lemma 5.6.5. *Fix $\epsilon,\delta\in(0,1)$. Let $\boldsymbol{W}^\star\in\mathbb{R}^{k\times d}$ be the true parameter matrix. There exists a matrix $\widetilde{\boldsymbol{W}}^\star\in\mathbb{R}^{k\times d}$ such that, with probability at least $1-\delta$:*

- $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}((\boldsymbol{W}_i^\star-\boldsymbol{W}_j^\star)\cdot\boldsymbol{x})\neq\mathrm{sgn}((\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star)\cdot\boldsymbol{x})]\le\epsilon$ *for all $i\neq j$, and,*
- $\|\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star\|_2\ge 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ *for any $i\neq j$.*

Proof of Lemma 5.6.5. The above lemma is a result of the next Section 5.6.2. In particular, it is a direct implication of Lemma 5.6.7 and Corollary 5.6.9. □

Note that the above lemma implies that

$$(\forall i,j)\qquad \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{v}_{ij}\cdot\boldsymbol{x})\neq\mathrm{sgn}((\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star)\cdot\boldsymbol{x})]\le 2\epsilon\,,$$

with probability at least $1-2\delta$. Hence, up to constants, the analysis concerning the feasibility of the true matrix $\boldsymbol{W}^\star$ (see Item 1) will still hold for $\widetilde{\boldsymbol{W}}^\star$. From now on we can work with this matrix $\widetilde{\boldsymbol{W}}^\star$ which enjoys the "well-conditionedness" property of the second item of the lemma.

We will use the above lemma in order to prove Item 3 which controls the volume of the feasible region: it states that there exist $0<r<R$ so that the feasible region of the convex program contains a ball of radius r and is contained in a ball of radius R (where the balls are with respect to the Frobenius norm). Moreover, $r=2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ and $R=1$.

For the chosen $\phi\in(0,1)$, the feasible set contains matrices $\boldsymbol{W}\in\mathbb{R}^{k\times d}$ that satisfy $\|\boldsymbol{W}-\widetilde{\boldsymbol{W}}^\star\|_F\le 2r$, r to be decided. For any $i\neq j$, we have that the following properties hold:

1. $\|\widetilde{\boldsymbol{W}}_i^\star-\widetilde{\boldsymbol{W}}_j^\star\|_2\ge 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ (well-conditionedness).

2. $(\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star) \cdot \boldsymbol{v}_{ij} \geq (1-\phi) \, \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2$ (feasibility).

3. $\|\boldsymbol{W} - \widetilde{\boldsymbol{W}}^\star\|_F \leq 2r$ which implies that $\|\boldsymbol{W}_i - \widetilde{\boldsymbol{W}}_i^\star\|_2 \leq 2r$ for any $i \in [k]$ (ball around feasible point).

4. $\|\boldsymbol{v}_{ij}\|_2 = 1$.

Our goal is to prove that for a matrix in the above ball it holds $(\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} \geq (1-\phi) \, \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2$.

We have that

$$\begin{aligned}(\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star) \cdot \boldsymbol{v}_{ij} &= (\widetilde{\boldsymbol{W}}_i^\star - \boldsymbol{W}_i) \cdot \boldsymbol{v}_{ij} + (\boldsymbol{W}_j - \widetilde{\boldsymbol{W}}_j^\star) \cdot \boldsymbol{v}_{ij} + (\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} \\ &\leq \|\widetilde{\boldsymbol{W}}_i^\star - \boldsymbol{W}_i\|_2 + \|\boldsymbol{W}_j - \widetilde{\boldsymbol{W}}_j^\star\|_2 + (\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} \\ &\leq 4r + (\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} \, .\end{aligned}$$

More to that

$$\begin{aligned}\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 &= \|\boldsymbol{W}_i - \widetilde{\boldsymbol{W}}_i^\star + \widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star + \widetilde{\boldsymbol{W}}_j^\star - \boldsymbol{W}_j\|_2 \\ &\leq \|\boldsymbol{W}_i - \widetilde{\boldsymbol{W}}_i^\star\|_2 + \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2 + \|\widetilde{\boldsymbol{W}}_j^\star - \boldsymbol{W}_j\|_2 \\ &\leq 4r + \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2 \, ,\end{aligned}$$

and similarly: $\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 \geq \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2 - 4r$.

Combining the above inequalities, we get that

$$\begin{aligned}(\boldsymbol{W}_i - \boldsymbol{W}_j) \cdot \boldsymbol{v}_{ij} &\geq (\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star) \cdot \boldsymbol{v}_{ij} - 4r \\ &\geq (1-\phi) \, \|\widetilde{\boldsymbol{W}}_i^\star - \widetilde{\boldsymbol{W}}_j^\star\|_2 - 4r \\ &\geq (1-\phi) \, (\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 - 4r) - 4r \\ &= (1-\phi) \, \|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 - 8r \, .\end{aligned}$$

We pick r sufficiently small and of order $2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ and get that $\boldsymbol{W}$ is a feasible solution of the convex program. Moreover, we can select $R = 1$ since $\|\widetilde{\boldsymbol{W}}^\star\|_F = 1$ without loss of generality, since we can normalize the row differences of $\widetilde{\boldsymbol{W}}^\star$ with the norm $\|\widetilde{\boldsymbol{W}}^\star\|_F$.

Proof of Item 4. We apply the ellipsoid algorithm in order to solve the convex program 5.1 and compute a matrix $\widetilde{\boldsymbol{W}} \in \mathbb{R}^{k \times d}$. The algorithm `ProperLSF` outputs the linear sorting function $h(\cdot) = \sigma_{\widetilde{\boldsymbol{W}}}(\cdot)$.

Lemma 5.6.6 (Efficiency of the Ellipsoid Algorithm (Vis21))**.** *Suppose that $P \subseteq \mathbb{R}^d$ is a full-dimensional polytope that is contained in a d-dimensional Euclidean ball of radius $R > 0$ and contains a d-dimensional Euclidean ball of radius $r > 0$. Then, the ellipsoid method outputs a point $\widetilde{\boldsymbol{x}} \in P$ after $O(d^2 \log(R/r))$ iterations. Moreover, every iteration can be implemented in $O(d^2 + T_{\mathrm{sep}})$ time, where T_{sep} is the time required to answer a single query by the separation oracle.*

Assume that Item 3 holds true. Then the algorithm can be used with $r = 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ and $R = 1$. Hence, the ellipsoid algorithm will provide in time $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$ a point $\widetilde{\boldsymbol{W}}$ that lies in the feasible region of the convex program 5.1[2].

□

Remark 5. *We remark that both the improper (Algorithm 6) and the proper (Algorithm 7) learning algorithms hold for the more general case where the $\boldsymbol{x}$-marginal lies in the class of isotropic log-concave distributions (LV07): A distribution $\mathcal{D}_x$ lies inside the class of isotropic log-concave distributions $\mathcal{F}_{\mathrm{LC}}$ over $\mathbb{R}^d$ if $\mathcal{D}_x$ has a probability density function f over $\mathbb{R}^d$ such that $\log f$ is concave, its mean is zero, and its covariance is identity, i.e., $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_x}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{I}$.*

The proof of Lemma 5.6.5

We provide the following result.

Lemma 5.6.7. *Fix $\epsilon, \delta \in (0, 1)$. Let $\boldsymbol{W}^\star \in \mathbb{R}^{k\times d}$ be the true parameter matrix. There exists a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ such that, with probability at least $1 - \delta$:*

- $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)\cdot\boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i - \boldsymbol{W}_j)\cdot\boldsymbol{x})] \leq \epsilon$ *for all $i \neq j$, and,*
- *The bit complexity of $\boldsymbol{W}$ is* $\mathrm{poly}(k, d, 1/\epsilon, \log(1/\delta))$

Proof. The matrix $\boldsymbol{W}$ will be the output of a linear program that can be used to learn the LSF $\sigma_{\boldsymbol{W}^\star}(\cdot)$ in the noiseless setting.

Consider the unit sphere $\mathcal{S}^{d-1}$ and a δ_0-cover of the unit sphere with parameter $\delta_0 > 0$ to be decided. For any sample $(\boldsymbol{x}, \pi) \sim \mathcal{D}$ of the 0-noisy linear label ranking distribution, i.e., $\boldsymbol{x} \sim \mathcal{N}_d$ and $\pi = \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$, we consider the rounded sample $(\widetilde{\boldsymbol{x}}, \pi)$ where $\widetilde{\boldsymbol{x}}$ is obtained by first projecting $\boldsymbol{x} \in \mathbb{R}^d$ to $\mathcal{S}^{d-1}$ and then by obtaining the closest point of $\widehat{x}$ in the cover. The cover's size is $O(1/\delta_0)^d$.

Let us fix $1 \leq i < j \leq k$ and set $y_{ij} = \mathrm{sgn}(\pi(i) - \pi(j))$. For a training set $\{(\boldsymbol{x}^{(t)}, \pi^{(t)})\}_{t\in[N]}$ of size N, we create the following linear system L_{ij} with variables $\boldsymbol{W} \in \mathbb{R}^{k\times d}$:

$$y_{ij}^{(t)}\ (\boldsymbol{W}_i - \boldsymbol{W}_j)\cdot\widetilde{\boldsymbol{x}}^{(t)} \geq 0\,,\ t \in [N] \qquad (\mathrm{L}_{ij})\,.$$

Consider the concatenation of the linear systems $\mathrm{L} = \cup_{i<j}\mathrm{L}_{ij}$. The number of equations in the linear system of equations L is $N \cdot \binom{k}{2}$.

We first have to show that, with high probability, the system L is feasible, i.e., there exists $\boldsymbol{W}$ that satisfies the system's equations. Note that if we replace $\widetilde{\boldsymbol{x}}^{(t)}$ with the original points $\boldsymbol{x}^{(t)}$, the true matrix $\boldsymbol{W}^\star$ is a solution to the system. We now have to study the rounded linear system.

[2] We remark that the runtime will also depend on the time required to answer a single query by the separation oracle. We assume that this time is polynomial in the parameters of our problem and we opt not to track these details in our work.

Claim 13. *The (rounded) linear system* L *is feasible with high probability.*

Proof. In order to show the feasibility of L, we will use the anti-concentration properties of the Gaussian.

Fact 4 ((DKM05))**.** *Let $\mathcal{P}$ be the standard normal distribution over $\mathbb{R}^d$. For any fixed unit vector $\boldsymbol{a} \in \mathbb{R}^d$ and any $\gamma \leq 1$,*

$$\gamma/4 \leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{P}}\left[|\boldsymbol{a} \cdot \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}| \leq \frac{\gamma}{\sqrt{d}}\right] \leq \gamma\,.$$

Let us focus on the pair $1 \leq i < j \leq k$. We first observe that scaling all samples to lie on the unit sphere does not affect the feasibility of the system. It suffices to focus on that single halfspace with normal vector $\boldsymbol{v}_{ij} = \boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star \in \mathbb{R}^d$ and consider the probability of the event that the collection of the N rounded points $\{\widetilde{\boldsymbol{x}}^{(t)}\}_t$ with labels $\{y_{ij}^{(t)}\}_t$, that come from N Gaussian vectors $\{\boldsymbol{x}^{(t)}\}_t$ which are linearly separable (with labels $\{y_{ij}^{(t)}\}_t$), becomes non-linearly separable. For this it suffices to control the probability that the rounding procedure flips the label of the data point. Using the union bound, we have that, if the rounding has accuracy δ_0, the described bad event has probability

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(N)}\sim\mathcal{N}_d}[\exists t \in [N] : \mathrm{sgn}(\boldsymbol{v}_{ij}\cdot\widetilde{\boldsymbol{x}}^{(t)}) \neq \mathrm{sgn}(\boldsymbol{v}_{ij}\cdot\boldsymbol{x}^{(t)})] \leq N\cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[|\boldsymbol{v}_{ij} \cdot \boldsymbol{x}/\|\boldsymbol{x}\|_2| \leq 2\delta_0]\ ,$$

and,

$$N \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[|\boldsymbol{v}_{ij} \cdot \boldsymbol{x}/\|\boldsymbol{x}\|_2| \leq 2\delta_0] \leq N \cdot O(\delta_0\sqrt{d})\,,$$

where we remark that the first event is scale invariant and so we can assume that the normal vector is unit, the first inequality follows from the fact that it suffices to control the mass assigned to a strip of width $2\delta_0$ (due to the discretization) and the second inequality follows from Fact 4. We now have to select the discretization. Let $\delta \in (0,1)$. By choosing $\delta_0 = O(\frac{\delta}{N\sqrt{d}k^2})$, the bad event for all the pairs $i < j$ occurs with probability at most δ, i.e., with probability at least $1-\delta$, each one of the N drawn i.i.d. samples does not fall in any one of the $\binom{k}{2}$ "bad" strips. □

We can now consider the case that the system L is feasible (with the target matrix $\boldsymbol{W}^\star$ being a feasible point) that occurs with probability $1-\delta$. The class of homogenous halfspaces in d dimensions has VC dimension d; therefore, the sample complexity of learning halfspaces using ERM is $O((d+\log(1/\delta))/\epsilon)$. Moreover, in the realizable case, we can implement the ERM using e.g., linear programming and find a solution in $\mathrm{poly}(d, 1/\epsilon, \log(1/\delta))$ time. We next focus on the quality of the solution which will give the desired sample complexity.

Claim 14. *Assume that the algorithm draws $N = \widetilde{O}(\frac{d+\log(k/\delta)}{\epsilon})$ i.i.d. samples of the form $(\boldsymbol{x}, \pi)$ with $\boldsymbol{x} \sim \mathcal{N}_d$ and $\pi = \sigma_{\boldsymbol{W}^\star}(\boldsymbol{x})$. For any $i \neq j$ and with probability at least $1-2\delta$, the solution $\boldsymbol{W}$ of the linear system* L *satisfies*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}((\boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)\cdot\boldsymbol{x}) \neq \mathrm{sgn}((\boldsymbol{W}_i - \boldsymbol{W}_j)\cdot\boldsymbol{x})] \leq \epsilon\,.$$

Proof. Since the matrix $\boldsymbol{W}$ satisfies the sub-system L_{ij}, the result follows using a union bound on the events that (i) the linear system is feasible and (ii) the ERM is a successful PAC learner. □

Claim 15. *Consider the solution $\boldsymbol{W}$ of the linear system. Then, $\boldsymbol{W}$ has bounded bit complexity of order* $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$.

Proof. We will make use of the following result that relates the size of the input and the output of a linear program using Cramer's rule.

Lemma 5.6.8 ((Sch98; Pap81))**.** *Let $\boldsymbol{A} \in \mathbb{Z}^{m\times n}, \boldsymbol{b} \in \mathbb{Z}^m, \boldsymbol{c} \in \mathbb{Z}^n$. Consider a linear program $\min \boldsymbol{c}\cdot\boldsymbol{x}$ subject to $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$ and $\boldsymbol{x} \ge \boldsymbol{0}$. Let U be the maximum size of A_{ij}, b_i, c_j. The output of the linear program has size $O(m(nU + n\log(n)))$ bits.*

We will apply the above lemma (which holds even by dropping the constraint $\boldsymbol{x} \ge \boldsymbol{0}$) to our setting where $\boldsymbol{A}\boldsymbol{w} \ge 0$ where $\boldsymbol{w} = (\boldsymbol{W}_i)_{i\in[k]} \in \mathbb{Q}^{kd}$, i.e., $\boldsymbol{w}$ is the vectorization of the matrix $\boldsymbol{W}$. Moreover, $\boldsymbol{A}$ is the matrix containing the N (rounded) Gaussian samples $\widetilde{\boldsymbol{x}}^{(t)}$. We have that the matrix $\boldsymbol{A}$ has dimension $N\binom{k}{2} \times kd$ and each entry A_{ij} is an integer and has size at most $U = \mathrm{poly}(d, k)$ (since the samples are rounded on the δ_0-cover of the sphere. Recall that the labels $y_{ij}^{(t)} \in \{-1, +1\}$ and $\widetilde{\boldsymbol{x}}^{(t)}$ lie in the unit sphere. In particular, each row of the matrix $\boldsymbol{A}$ has $2d$ non-zero entries and is associated with a tuple (i, j, t) for $1 \le i < j \le k$ and $t \in [N]$. Then, it holds that the output has size at most $O(Nk^2(dU + dk\log(dk)))$ bits. So, we get that the output $\boldsymbol{W}$ can be described using at most $\mathrm{poly}(d, k, 1/\epsilon, U, \log(1/\delta)) = \mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$ bits (due to the size of the entries of the matrix $\boldsymbol{A}$). □

Combining the above claims, we conclude the proof. □

As a corollary of the bounded bit complexity, we obtain the following key result.

Corollary 5.6.9. *Let $\epsilon > 0$. Assume that $\boldsymbol{W} \in \mathbb{R}^{k\times d}$ has bit complexity at most* $\mathrm{poly}(d, k, 1/\epsilon, \log(1/\delta))$. *Then, for any $i, j \in [k]$ with $i \neq j$, it holds that $\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 > 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$.*

Proof. First, we can assume that $\boldsymbol{W}_i \neq \boldsymbol{W}_j$ for any $i \neq j$; in case of equal rows, we obtain a low-dimensional instance. Then, since any vector $\boldsymbol{W}_i$ has bounded bit complexity, we have that the difference of any two such vectors, provided that it is non-zero, has a lower bound in its norm, i.e., $\|\boldsymbol{W}_i - \boldsymbol{W}_j\|_2 > 2^{-\mathrm{poly}(d,k,1/\epsilon,\log(1/\delta))}$ for any $i, j \in [k]$. □

5.7 Learning in Top-1 Disagreement from Label Rankings

Let us set $\sigma_1(\boldsymbol{W}\boldsymbol{x}) = \mathrm{argmax}_{i\in[k]} \boldsymbol{W}_i \cdot \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{R}^d$. The main result of this section follows.

Theorem 5.7.1 (Proper Top-1 Learning Algorithm)**.** *Fix $\eta \in [0, 1/2)$ and $\epsilon, \delta \in (0, 1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1. There exists an algorithm that draws $N = O\left(\frac{dk\sqrt{\log k}}{\epsilon(1-2\eta)^6}\log(k/\delta)\right)$ samples from $\mathcal{D}$, runs in $\mathrm{poly}(N)$ time and, with probability at least $1-\delta$, outputs a Linear Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$ that is ϵ-close in top-1 disagreement to the target.*

Proof. Note that the `MassartLTF` algorithm (see Lemma 5.6.2) has the guarantee that it returns a vector $\boldsymbol{w}$ so that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\mathrm{sgn}(\boldsymbol{w}\cdot\boldsymbol{x}) \neq \mathrm{sgn}(\boldsymbol{w}^\star\cdot\boldsymbol{x})] \leq \epsilon\,,$$

with probability $1-\delta$, where $\boldsymbol{w}^\star$ is the target normal vector. Since the above misclassification probability with respect to $\mathcal{N}_d$ is directly connected with the angle $\theta(\boldsymbol{w}, \boldsymbol{w}^\star)$, we get that we can control the angle between $\boldsymbol{w}$ and $\boldsymbol{w}^\star$ efficiently. Moreover, in our setting, for a matrix $\boldsymbol{W} \in \mathbb{R}^{k\times d}$, there exist $\binom{k}{2}$ homogeneous halfspaces with normal vectors $\boldsymbol{W}_i - \boldsymbol{W}_j$ and so we can control the angles $\theta(\boldsymbol{W}_i - \boldsymbol{W}_j, \boldsymbol{W}_i^\star - \boldsymbol{W}_j^\star)$. In order to deduce the sample complexity bound of Theorem 5.7.1, we show the next lemma which essentially bounds the top-1 misclassification error using the angles of these $O(k^2)$ halfspaces. We apply Lemma 5.7.2 with $\boldsymbol{U} = \boldsymbol{W}$ and $\boldsymbol{V} = \boldsymbol{W}^\star$ and so we can take $\epsilon' = \epsilon/(k\sqrt{\log k})$ and invoke the proper learning algorithm of Algorithm 7. This completes the proof. □

We continue with the proof of our key lemma.

Lemma 5.7.2 (Misclassification Error)**.** *Consider two matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{k\times d}$ and let $\mathcal{N}_d$ be the standard Gaussian in d dimensions. We have that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] \leq c\cdot k\cdot\sqrt{\log k}\cdot\max_{i\neq j}\theta(\boldsymbol{U}_i - \boldsymbol{U}_j, \boldsymbol{V}_i - \boldsymbol{V}_j)\,,$$

where $c > 0$ is some universal constant.

Proof. We have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_1(\boldsymbol{V}\boldsymbol{x})] = \sum_{i\in[k]}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) = i, \sigma_1(\boldsymbol{V}\boldsymbol{x}) \neq i]\,.$$

We have that $\mathcal{C}_{\boldsymbol{U}}^{(i)} = \mathbf{1}\{\boldsymbol{x} : \sigma_1(\boldsymbol{U}\boldsymbol{x}) = i\} = \prod_{j\neq i}\mathbf{1}\{(\boldsymbol{U}_i - \boldsymbol{U}_j)\cdot\boldsymbol{x} \geq 0\}$ is the set indicator of a homogeneous polyhedral cone as the intersection of $k-1$ homogeneous halfspaces. Similarly, we consider the cone $\mathcal{C}_{\boldsymbol{V}}^{(i)} = \{\boldsymbol{x} : \sigma_1(\boldsymbol{V}\boldsymbol{x}) = i\}$. Hence, we have that $\{\boldsymbol{x} : \sigma_1(\boldsymbol{V}\boldsymbol{x}) \neq i\}$ is the complement of a homogeneous polyhedral cone. Let us define $C_{\boldsymbol{U}}^{(i)} : \mathbb{R}^d \mapsto \{0, 1\}$ and $C_{\boldsymbol{V}}^{(i)} : \mathbb{R}^d \mapsto \{0, 1\}$ be the associated indicator functions of the two cones. We have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_1(\boldsymbol{U}\boldsymbol{x}) = i, \sigma_1(\boldsymbol{V}\boldsymbol{x}) \neq i] = \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_{\boldsymbol{U}}^{(i)}(\boldsymbol{x}) = 1, C_{\boldsymbol{V}}^{(i)}(\boldsymbol{x}) = 0]\,.$$

Finally, we have that

$$\mathcal{C}_{\boldsymbol{U}}^{(i)} \cap \left(\mathcal{C}_{\boldsymbol{V}}^{(i)}\right)^c = \mathcal{C}_{\boldsymbol{U}}^{(i)} \setminus \mathcal{C}_{\boldsymbol{V}}^{(i)} \subseteq \mathcal{C}_{\boldsymbol{U}}^{(i)} \setminus \mathcal{C}_{\boldsymbol{V}}^{(i)} \cup \mathcal{C}_{\boldsymbol{V}}^{(i)} \setminus \mathcal{C}_{\boldsymbol{U}}^{(i)} .$$

We can hence apply Lemma 5.7.3 for the cones $\mathcal{C}_{\boldsymbol{U}}^{(i)}, \mathcal{C}_{\boldsymbol{V}}^{(i)}$ for each $i \in [k]$. □

Lemma 5.7.3 (Cone Disagreement). *Let $C_1 : \mathbb{R}^d \mapsto \{0,1\}$ be the indicator function of the homogeneous polyhedral cone defined by the k unit vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{R}^d$, i.e., $C_1(\boldsymbol{x}) = \prod_{i=1}^k \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \geq 0\}$. Similarly, define $C_2 : \mathbb{R}^d \mapsto \{0,1\}$ to be the homogeneous polyhedral cone with normal vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$. It holds that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq c\sqrt{\log(k)}\ \max_{i \in [k]} \theta(\boldsymbol{v}_i, \boldsymbol{u}_i)\,,$$

where $c > 0$ is some universal constant.

Proof. To simplify notation, denote $\theta = \max_{i \in [k]} \theta(\boldsymbol{v}_i, \boldsymbol{u}_i)$. We first observe that it suffices to prove the upper bound on the probability of $C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})$ for sufficiently small values of θ. Indeed, if we have that the bound is true for θ smaller than some θ_0 we can then form a path of sufficiently large length N (in particular we need $\theta/N \leq \theta_0$) starting from the vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ to the final vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$, where at each step we only rotate the vectors by at most $\theta/N \leq \theta_0$. By the triangle inequality, we immediately obtain that the probability that $C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})$ is at most equal to the sum of the probabilities of the intermediate steps which is at most $\sum_{i=1}^N c\sqrt{\log(k)}\frac{\theta}{N} = c\sqrt{\log(k)}\theta$. Notice in the above argument the constant θ_0 can be arbitrarily small and may also depend on k and d.

We define the indicator of the positive orthant in k dimensions to be $R(\boldsymbol{t}) = \prod_{i=1}^k \mathbf{1}\{\boldsymbol{t}_i \geq 0\}$. Using this notation, we have that the cone indicator can be written as $C_1(\boldsymbol{x}) = R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) = R(\boldsymbol{V}\boldsymbol{x})$, where $\boldsymbol{V}$ is the $k \times d$ matrix whose i-th row is the vector $\boldsymbol{v}_i$. Moreover, we define the i-th face of the cone $R(\boldsymbol{V}\boldsymbol{x})$ to be

$$F_i(\boldsymbol{V}\boldsymbol{x}) = R(\boldsymbol{V}\boldsymbol{x})\ \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} = 0\}\,.$$

We will first handle the case where only one of the normal vectors $\boldsymbol{v}_i$ changes. We show the following claim.

Claim 16. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{R}^d$ and $\boldsymbol{r} \in \mathbb{R}^d$ with $\theta(\boldsymbol{v}_1, \boldsymbol{r}) \leq \theta$ for some sufficiently small $\theta \in (0, \pi/2)$. It holds that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[R(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq R(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})] \leq c \cdot \theta \cdot \Gamma(F_1)\sqrt{\log\left(\frac{1}{\Gamma(F_1)} + 1\right)}\,,$$

where F_1 is the face with $\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0$ of the cone $R(\boldsymbol{V}\boldsymbol{x})$ and c is some universal constant.

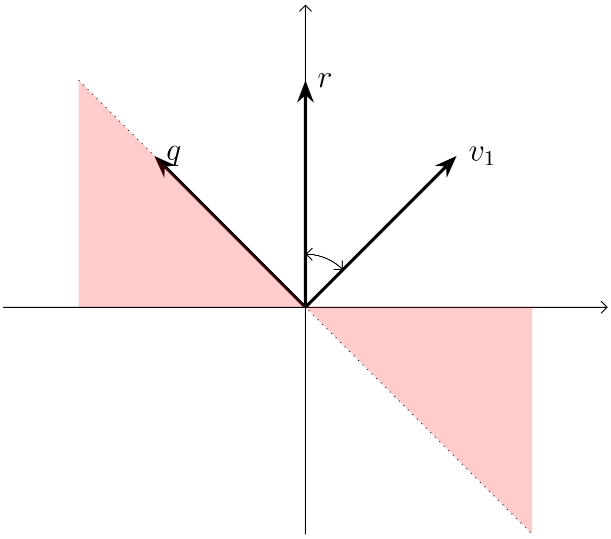


Figure 5.1: The vectors $\boldsymbol{r}, \boldsymbol{v}_1$ and $\boldsymbol{q}$ and the disagreement region of the halfspaces with normal vectors $\boldsymbol{r}$ and $\boldsymbol{v}_1$.

Proof. We have

$$\begin{aligned}
&\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x}) \neq R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\right] \\
&\quad= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[|R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x}) - R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})|\right] \\
&\quad= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ |\mathbf{1}\{\boldsymbol{v}_1\cdot\boldsymbol{x}\geq 0\} - \mathbf{1}\{\boldsymbol{r}\cdot\boldsymbol{x}\geq 0\}|\right] .
\end{aligned}$$

We have that $|\mathbf{1}\{\boldsymbol{v}_1\cdot\boldsymbol{x}\geq 0\} - \mathbf{1}\{\boldsymbol{r}\cdot\boldsymbol{x}\geq 0\}| = \mathbf{1}\{(\boldsymbol{v}_1\cdot\boldsymbol{x})(\boldsymbol{r}\cdot\boldsymbol{x}) < 0\}$, i.e., this is the event that the halfspaces $\mathbf{1}\{\boldsymbol{v}_1\cdot\boldsymbol{x}\geq 0\}$ and $\mathbf{1}\{\boldsymbol{r}\cdot\boldsymbol{x}\geq 0\}$ disagree. Let $\boldsymbol{q}$ be the normalized projection of $\boldsymbol{r}$ onto the orthogonal complement of $\boldsymbol{v}_1$, i.e., $\boldsymbol{q} = \mathrm{proj}_{\boldsymbol{v}_1^\perp}\boldsymbol{r}/\|\mathrm{proj}_{\boldsymbol{v}_1^\perp}\boldsymbol{r}\|_2$. We have that $\boldsymbol{v}_1$ and $\boldsymbol{q}$ is an orthonormal basis of the subspace spanned by the vectors $\boldsymbol{v}_1$ and $\boldsymbol{r}$. We have that $\boldsymbol{r} = \cos\theta(\boldsymbol{v}_1,\boldsymbol{r})\boldsymbol{v}_1 + \sin\theta(\boldsymbol{v}_1,\boldsymbol{r})\boldsymbol{q}$. Moreover, we have that the region $(\boldsymbol{v}_1\cdot\boldsymbol{x})(\boldsymbol{r}\cdot\boldsymbol{x}) < 0$ is equal to

$$\{0 < \boldsymbol{v}_1\cdot\boldsymbol{x} < -(\boldsymbol{q}\cdot\boldsymbol{x})\tan\theta(\boldsymbol{v}_1,\boldsymbol{r})\} \cup \{-(\boldsymbol{q}\cdot\boldsymbol{x})\tan\theta(\boldsymbol{v}_1,\boldsymbol{r}) < \boldsymbol{v}_1\cdot\boldsymbol{x} < 0\}\,.$$

Thus, we have that the disagreement region $(\boldsymbol{v}_1\cdot\boldsymbol{x})(\boldsymbol{r}\cdot\boldsymbol{x}) < 0$ is a subset of the region $\{|\boldsymbol{v}_1\cdot\boldsymbol{x}| \leq |\boldsymbol{q}\cdot\boldsymbol{x}|\tan\theta(\boldsymbol{v}_1,\boldsymbol{r})\}$. Since $\tan\theta(\boldsymbol{v}_1,\boldsymbol{r}) \leq \theta$ and we have that θ is sufficiently small we can also replace the above region by the larger region: $\{|\boldsymbol{v}_1\cdot\boldsymbol{x}| \leq 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}$. Therefore, we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ \mathbf{1}\{(\boldsymbol{v}_1\cdot\boldsymbol{x})(\boldsymbol{r}\cdot\boldsymbol{x}) < 0\}\right] \\
&\qquad\leq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ \mathbf{1}\{|\boldsymbol{v}_1\cdot\boldsymbol{x}| \leq 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}\right] .
\end{aligned}$$

The derivative of the above expression with respect to θ is equal to

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ \delta\left(\frac{|\boldsymbol{v}_1\cdot\boldsymbol{x}|}{2|\boldsymbol{q}\cdot\boldsymbol{x}|}-\theta\right)\right],$$

where $\delta(t)$ is the Dirac delta function. At $\theta = 0$ and using the property that $\delta(t/a) = a\delta(t)$, we have that the above derivative is equal to

$$2\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ |\boldsymbol{q}\cdot\boldsymbol{x}|\ \delta(|\boldsymbol{v}_1\cdot\boldsymbol{x}|)\right].$$

Notice that, if we did not have the term $|\boldsymbol{q}\cdot\boldsymbol{x}|$, the above expression would be exactly equal to two times the Gaussian surface area of the face with $\boldsymbol{v}_1\cdot\boldsymbol{x} = 0$, i.e., it would be equal to $2\Gamma(F_1)$. We now show that this extra term of $|\boldsymbol{q}\cdot\boldsymbol{x}|$ can only increase the above surface integral by at most a logarithmic factor. We have that

$$\begin{aligned}&\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ |\boldsymbol{q}\cdot\boldsymbol{x}|\ \delta(|\boldsymbol{v}_1\cdot\boldsymbol{x}|)\right] = \int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|d\mu(\boldsymbol{x})\\
&\le \int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\le\xi\}d\mu(\boldsymbol{x}) + \int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\ge\xi\}d\mu(\boldsymbol{x})\\
&\le \xi\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})d\mu(\boldsymbol{x}) + \int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\ge\xi\}d\mu(\boldsymbol{x}),\end{aligned}$$

where $d\mu(\boldsymbol{x})$ is the standard surface measure in $\mathbb{R}^d$. The first term above is exactly equal to the Gaussian surface area of the face F_1. To bound from above the second term we can use the fact that the face F_1 is a subset of the hyperplane $\boldsymbol{v}_1\cdot\boldsymbol{x} = 0$, i.e., it holds that $F_1 \subseteq \{\boldsymbol{x} : |\boldsymbol{v}_1\cdot\boldsymbol{x}| = 0\}$. To simplify notation we may assume that $\boldsymbol{v}_1 = \boldsymbol{e}_1$ and $\boldsymbol{q} = \boldsymbol{e}_2$ (recall that $\boldsymbol{v}_1$ and $\boldsymbol{q}$ are orthogonal unit vectors), and in this case we obtain

$$\begin{aligned}\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q}\cdot\boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q}\cdot\boldsymbol{x}|\ge\xi\}d\mu(\boldsymbol{x}) &\le \int_{x_1=0}\phi_d(\boldsymbol{x})|x_2|\mathbf{1}\{|x_2|\ge\xi\}d\mu(\boldsymbol{x})\\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty}|x_2|\mathbf{1}\{|x_2|\ge\xi\}\frac{e^{-x_2^2/2}}{\sqrt{2\pi}}dx_2\\
&= \frac{1}{\pi}e^{-\xi^2/2}.\end{aligned}$$

Combining the above bounds we obtain that the derivative with respect to θ of the expression $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ \mathbf{1}\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\le 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}\right]$ is equal to

$$\frac{d}{d\theta}\Big(\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\ \mathbf{1}\{|\boldsymbol{v}_1\cdot\boldsymbol{x}|\le 2\theta|\boldsymbol{q}\cdot\boldsymbol{x}|\}\right]\Big)\Big|_{\theta=0} \le 2\xi\Gamma(F_1) + \frac{2e^{-\xi^2/2}}{\pi}.$$

By picking $\xi = \sqrt{2\log(1+1/\Gamma(F_1))}$, the result follows since up to introducing $o(\theta)$ error we can bound the term $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x}) \neq R(\boldsymbol{r}\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})\right]$ by its derivative with respect to θ (evaluated at 0) times θ. □

We can complete the proof of Lemma 5.7.3 using Claim 16. In order to bound the disagreement of the cones C_1 and C_2 we can start from C_1 and change one of its vectors at a time so that we can use Claim 16 that can handle this case. For example, at the first step, we can swap $\boldsymbol{v}_1$ for $\boldsymbol{u}_1$ and use the triangle inequality to obtain that

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq & \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{v}_1\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x}) \neq R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x}\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x})] \\ &+ \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x}) \neq R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{u}_2\cdot\boldsymbol{x}\ldots,\boldsymbol{u}_k\cdot\boldsymbol{x})] \\ &\leq c\cdot\theta\ \Gamma(F_1)\sqrt{\log(1/\Gamma(F_1)+1)} \\ &+ \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{v}_2\cdot\boldsymbol{x},\ldots,\boldsymbol{v}_k\cdot\boldsymbol{x}) \neq R(\boldsymbol{u}_1\cdot\boldsymbol{x},\boldsymbol{u}_2\cdot\boldsymbol{x}\ldots,\boldsymbol{u}_k\cdot\boldsymbol{x})],\end{aligned}$$

where $F_1 = F_1(\boldsymbol{V}\boldsymbol{x})$ is the face with $\boldsymbol{v}_1\cdot\boldsymbol{x} = 0$ of the cone C_1. Notice that we have replaced $\boldsymbol{v}_1$ by $\boldsymbol{u}_1$ in the above bound. Our plan is to use the triangle inequality and continue replacing the vectors of C_1 by the vectors of C_2 sequentially. To make this formal we define the matrix $\boldsymbol{A}^{(i)} \in \mathbb{R}^{k\times d}$ whose first $i-1$ rows are the vectors $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{i-1}$ and its last $k-i+1$ rows are the vectors $\boldsymbol{v}_i,\ldots,\boldsymbol{v}_k$, i.e.,

$$\boldsymbol{A}^{(i)}_j = \begin{cases}\boldsymbol{u}_j & \text{if} \quad 1\leq j\leq i-1, \\ \boldsymbol{v}_j & \text{if} \quad i\leq j\leq k\,.\end{cases}$$

Notice that $\boldsymbol{A}^{(1)} = \boldsymbol{V}$ and $\boldsymbol{A}^{(k+1)} = \boldsymbol{U}$. Using the triangle inequality we obtain that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq \sum_{i=1}^{k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[R(\boldsymbol{A}^{(i)}\boldsymbol{x}) \neq R(\boldsymbol{A}^{(i+1)}\boldsymbol{x})].$$

Since the matrices $\boldsymbol{A}^{(i)}$ and $\boldsymbol{A}^{(i+1)}$ only differ on one row, we can use Claim 16 to obtain the following bound:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq c\cdot\theta\cdot\sum_{i=1}^{k}\Gamma(F_i(\boldsymbol{A}^{(i)}\boldsymbol{x}))\sqrt{\frac{1}{\Gamma(F_i(\boldsymbol{A}^{(i)}\boldsymbol{x}))}+1}\,.$$

We now observe that the Gaussian surface area $\Gamma(F_i(\boldsymbol{A}^{(i)}\boldsymbol{x}))$ is a continuous function of the matrix $\boldsymbol{A}^{(i)}$. By flattening the matrix $\boldsymbol{A}^{(i)}$ (since it is isomorphic to a vector $\boldsymbol{z} \in \mathbb{R}^{n^2}$) and letting $S_{\boldsymbol{z}}$ be the induced surface $\{\boldsymbol{x} : R(\boldsymbol{A}^{(i)}\boldsymbol{x}) = 1 \wedge \boldsymbol{v}_i\cdot\boldsymbol{x} = 0\}$, it suffices to show that

$$\lim_{\boldsymbol{w}\to\boldsymbol{z}}\int\phi_n(\boldsymbol{x})\mathbf{1}\{\boldsymbol{x}\in S_{\boldsymbol{w}}\}d\mu(\boldsymbol{x}) = \int\phi_n(\boldsymbol{x})\mathbf{1}\{\boldsymbol{x}\in S_{\boldsymbol{z}}\}d\mu(\boldsymbol{x})\,,$$

by the smoothness of the surface $S_{\boldsymbol{z}}$. Consider a sequence of functions (g_m) and vectors $(\boldsymbol{w}_m)$ so that $g_m(\boldsymbol{x}) = \phi_n(\boldsymbol{x})\mathbf{1}\{\boldsymbol{x}\in S_{\boldsymbol{w}_m}\}$ and $\lim_{m\to\infty}\boldsymbol{w}_m = \boldsymbol{z}$. Note that

$|g_m(\boldsymbol{x})| \leq 1$ everywhere. Hence, by the dominated convergence theorem, we have that

$$\lim_{m\to\infty} \int g_m(\boldsymbol{x}) d\mu(\boldsymbol{x}) = \int \lim_{m\to\infty} g_m(\boldsymbol{x}) d\mu(\boldsymbol{x}) = \int \phi_n(\boldsymbol{x}) \lim_{m\to\infty} \mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{w}_m}\} d\mu(\boldsymbol{x}) .$$

Since the sequence consists of smooth surfaces, we have that $\lim_{m\to\infty} \mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{w}_m}\} = \mathbf{1}\{\boldsymbol{x} \in S_{\boldsymbol{z}}\}$ and so the Gaussian surface area is continuous with respect to the matrix $\boldsymbol{A}^{(i)}$ for any $i \in [k]$.

Also, as $\theta \to 0$, we have that $\boldsymbol{A}^{(i)} \to \boldsymbol{V}$. This is because the sequence of matrices $\boldsymbol{A}^{(i)}$ depends only on the vectors $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ for $j \in [k]$ and the following two properties hold true: $\theta = \max_{j\in[k]} \theta(\boldsymbol{v}_j, \boldsymbol{u}_j)$ and all the vectors are unit. Hence, as θ tends to zero, they tend to become the same vectors and so any matrix $\boldsymbol{A}^{(i)}$ tends to become $\boldsymbol{V}$. Therefore, taking this limit we obtain that for $\theta \to 0$ it holds that

$$\lim_{\theta\to 0} \frac{\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})]}{\theta} \leq c \cdot \sum_{i=1}^{k} \Gamma(F_i(\boldsymbol{V}\boldsymbol{x}))\sqrt{\log\left(1/\Gamma(F_i(\boldsymbol{V}\boldsymbol{x})) + 1\right)} . \quad (5.1)$$

We will now use the following lemma that shows that the surface area of any homogeneous polyhedral cone is independent of the number of faces k and in fact is at most 1 for all k.

Lemma 5.7.4 (Gaussian Surface Area of Homogeneous Cones (Naz03))**.** *Let C be a cone with apex at the origin (i.e., an intersection of arbitrarily many halfspaces all of whose boundaries contain the origin). Then C has Gaussian surface area $\Gamma(C)$ at most* 1.

Using Lemma 5.7.4 we obtain that $\sum_{i=1}^{k} \Gamma(F_i(\boldsymbol{V}\boldsymbol{x})) \leq 1$. Next, we observe that, when the positive numbers $a_1, \ldots, a_k$ satisfy $\sum_{i=1}^{k} a_i \leq 1$, it holds that $\sum_{i=1}^{k} a_i \sqrt{\log(1/a_i)} \leq \sqrt{\sum_{i=1}^{k} a_i \log(1/a_i)} \leq \sqrt{\log(k)}$ (using the fact that the uniform distribution maximizes the entropy). Using this fact and Equation (5.1), we obtain

$$\lim_{\theta\to 0} \frac{\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})]}{\theta} \leq c\sqrt{\log(k)} .$$

Thus, we have shown that, for sufficiently small θ, it holds that $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[C_1(\boldsymbol{x}) \neq C_2(\boldsymbol{x})] \leq c\sqrt{\log(k)}\theta$, but, as we discussed in the start of the proof, the general bound follows directly from the bound for sufficiently small values of $\theta > 0$. □

5.8 Learning in Top-r Disagreement from Label Rankings

We prove the next result which corresponds to a proper learning algorithm for LSF in the presence of bounded noise with respect to the top-r disagreement.

Theorem 5.8.1 (Proper Top-r Learning Algorithm)**.** *Fix $\eta \in [0,1/2)$, $r \in [k]$ and $\epsilon, \delta \in (0,1)$. Let $\mathcal{D}$ be an η-noisy linear label ranking distribution satisfying the assumptions of Definition 5.1.1. There exists an algorithm that draws $N = \widetilde{O}\left(\frac{d\ rk}{\epsilon(1-2\eta)^6}\log(1/\delta)\right)$ samples from $\mathcal{D}$, runs in $\mathrm{poly}(N)$ time and, with probability at least $1-\delta$, outputs a Linear Sorting function $h : \mathbb{R}^d \to \mathbb{S}_k$ that is ϵ-close in top-r disagreement to the target.*

The main result of this section is the next lemma, which directly implies the above theorem (using the same steps as the proof of Theorem 5.7.1).

Lemma 5.8.2 (Top-r Misclassification)**.** *Let $r \in [k]$. Consider two matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{k\times d}$ and let $\mathcal{N}_d$ be the standard Gaussian in d dimensions. We have that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})] \leq c \cdot k \cdot r \cdot \sqrt{\log(kr)} \cdot \max_{i\neq j} \theta(\boldsymbol{U}_i - \boldsymbol{U}_j, \boldsymbol{V}_i - \boldsymbol{V}_j)\,,$$

where $c > 0$ is some universal constant.

Proof. Let us set $\sigma_{1..r}(\boldsymbol{W}\boldsymbol{x})$ denote the ordering of the top-r alternatives in the ranking $\sigma(\boldsymbol{W}\boldsymbol{x})$. Moreover, recall that $\sigma_\ell(\boldsymbol{W}\boldsymbol{x})$ denotes the alternative in the ℓ-th position of the ranking $\sigma(\boldsymbol{W}\boldsymbol{x})$. For two matrices $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{k\times d}$, we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})] = \sum_{j=1}^{k} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[\bigcup_{\ell=1}^{r}\{j = \sigma_\ell(\boldsymbol{U}\boldsymbol{x}), j \neq \sigma_\ell(\boldsymbol{V}\boldsymbol{x})\}\right].$$

The first step is to understand the geometry of the set $\bigcup_{\ell=1}^{r}\{\boldsymbol{x} : j = \sigma_\ell(\boldsymbol{U}\boldsymbol{x})\} = \{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{U}\boldsymbol{x})\}$ for $j \in [k]$. We have that this set is equal to

$$\mathcal{T}_{\boldsymbol{U}}^{(j)} = \bigcup_{S\subseteq[k]:|S|\leq r-1}\ \bigcap_{i\in S}\{\boldsymbol{x} : (\boldsymbol{U}_i - \boldsymbol{U}_j)\cdot\boldsymbol{x} \geq 0\} \cap \bigcap_{i\notin S}\{\boldsymbol{x} : (\boldsymbol{U}_i - \boldsymbol{U}_j)\cdot\boldsymbol{x} \leq 0\}\,.$$

In words, $\mathcal{T}_{\boldsymbol{U}}^{(j)}$ iterates over any possible collection of alternatives that can win the element j (they lie in the set of top elements S) and the remaining elements lose when compared with j (they lie in the complement set $[k]\setminus S$). Overloading the notation, let us define the mapping $T(\boldsymbol{t}) = T(t_1, ..., t_k) = \sum_{S\subseteq[k]:|S|\leq r-1}\prod_{i\in S}\mathbf{1}\{t_i \geq 0\}\prod_{i\notin S}\mathbf{1}\{t_i \leq 0\}$. Using this mapping, we can define the indicator of the set $T_{\boldsymbol{U}}^{(j)}$ as $T((\boldsymbol{U}_1 - \boldsymbol{U}_j)\cdot\boldsymbol{x}, \ldots, (\boldsymbol{U}_k - \boldsymbol{U}_j)\cdot\boldsymbol{x})$. The top-$r$ disagreement $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}), j \notin \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})]$ is equal to:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T((\boldsymbol{U}_1 - \boldsymbol{U}_j)\cdot\boldsymbol{x}, ..., (\boldsymbol{U}_k - \boldsymbol{U}_j)\cdot\boldsymbol{x}) = 1, T((\boldsymbol{V}_1 - \boldsymbol{V}_j)\cdot\boldsymbol{x}, ..., (\boldsymbol{V}_k - \boldsymbol{V}_j)\cdot\boldsymbol{x}) = 0]\,.$$

So we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})] = \sum_{j=1}^{k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x}) = 1, T_j(\boldsymbol{V}\boldsymbol{x}) = 0] \leq \sum_{j=1}^{k}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x}) \neq T_j(\boldsymbol{V}\boldsymbol{x})]\,.$$

In order to show the desired bound, it suffices to prove the following two lemmas.

Lemma 5.8.3 (Disagreement Region)**.** *Consider a positive integer $r \leq k$. Fix $j \in [k]$ and let $\theta = \max_{i \in [k]} \theta(\boldsymbol{U}_i - \boldsymbol{U}_j, \boldsymbol{V}_i - \boldsymbol{V}_j)$. Then it holds that*

$$\lim_{\theta \to 0} \frac{\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x}) \neq T_j(\boldsymbol{V}\boldsymbol{x})]}{\theta} \leq c \cdot \sum_{i \in [k]} \Gamma(F_i^j) \sqrt{\log\left(\frac{1}{\Gamma(F_i^j)} + 1\right)} ,$$

where $c > 0$ is some constant and F_i^j is the surface $\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})\} \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\}$ for the matrix $\boldsymbol{V} \in \mathbb{R}^{k \times d}$.

and,

Lemma 5.8.4. *Let F_i^j, r, k as in the previous lemma. It holds that*

$$\sum_{i \in [k]} \sum_{j \in [k]} \Gamma(F_i^j) \leq 2kr \,.$$

Applying these two lemmas with $\theta = \max_{i \neq j} \theta(\boldsymbol{U}_i - \boldsymbol{U}_j, \boldsymbol{V}_i - \boldsymbol{V}_j)$, we get that

$$Z := \lim_{\theta \to 0} \frac{\sum_{j \in [k]} \mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x}) \neq T_j(\boldsymbol{V}\boldsymbol{x})]}{\theta} \leq c \cdot \sum_{j \in [k]} \sum_{i \in [k]} \Gamma(F_i^j) \sqrt{\log\left(\frac{1}{\Gamma(F_i^j)} + 1\right)} .$$

Let us set $\Gamma'(F_i^j) = \Gamma(F_i^j)/(2kr)$. Then we have that

$$Z \leq 2ckr \cdot \sum_{j \in [k]} \sum_{i \in [k]} \Gamma'(F_i^j) \sqrt{\log\left(\frac{1}{2kr \cdot \Gamma'(F_i^j)} + 1\right)} .$$

It suffices to bound the quantity

$$\sum_{j \in [k]} \sum_{i \in [k]} \Gamma'(F_i^j) \sqrt{\log\left(\frac{1}{\Gamma'(F_i^j)} + 1\right)} = O\left(kr\sqrt{\log(kr)}\right) ,$$

where we used a similar "entropy-like" inequality as we did in the top-1 case. This yields (by recalling that it is sufficient to consider only the case of arbitrarily small angles, as in the top-1 case) that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[\sigma_{1..r}(\boldsymbol{U}\boldsymbol{x}) \neq \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})] \leq c\ rk\ \sqrt{\log(kr)} \cdot \max_{i \neq j} \theta(\boldsymbol{U}_i - \boldsymbol{U}_j, \boldsymbol{V}_i - \boldsymbol{V}_j) ,$$

for some universal constant c. □

5.8.1 The proof of Lemma 5.8.3

We proceed with the proof of the key lemma concerning the disagreement region. We first show the following claim where we only change a single vector. Recall that

$$T(\boldsymbol{V}\boldsymbol{x}) = \sum_{S:|S|\le r-1} \prod_{i\in S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \ge 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \le 0\}\,.$$

We will be interested in the surface $F_1 := F_1(\boldsymbol{V}\boldsymbol{x}) = T(\boldsymbol{V}\boldsymbol{x})\mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0\}$.

Claim 17. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{R}^d$ and $\boldsymbol{r} \in \mathbb{R}^d$ with $\theta(\boldsymbol{v}_1, \boldsymbol{r}) \le \theta$ for some sufficiently small $\theta \in (0, \pi/2)$. It holds that*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T(\boldsymbol{v}_1\cdot\boldsymbol{x}, \ldots, \boldsymbol{v}_k\cdot\boldsymbol{x}) \ne T(\boldsymbol{r}\cdot\boldsymbol{x}, \boldsymbol{v}_2\cdot\boldsymbol{x}, \ldots, \boldsymbol{v}_k\cdot\boldsymbol{x})] \le c\cdot\theta\cdot\Gamma(F_1)\sqrt{\log\left(\frac{1}{\Gamma(F_1)}+1\right)}\,,$$

where F_1 is the surface $T(\boldsymbol{V}\boldsymbol{x}) \cap \{\boldsymbol{x} : \boldsymbol{v}_1 \cdot \boldsymbol{x} = 0\}$ and c is some universal constant.

Proof. We first decompose the sum of $T(\boldsymbol{V}\boldsymbol{x})$ depending on whether $1 \in S$ or not. Hence, we have that $T(\boldsymbol{v}_1\cdot\boldsymbol{x}, \ldots, \boldsymbol{v}_k\cdot\boldsymbol{x}) = T^+(\boldsymbol{v}_1\cdot\boldsymbol{x}, \ldots, \boldsymbol{v}_k\cdot\boldsymbol{x}) + T^-(\boldsymbol{v}_1\cdot\boldsymbol{x}, \ldots, \boldsymbol{v}_k\cdot\boldsymbol{x})$ where

$$\begin{aligned}
T^+(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) &= \sum_{S\subseteq[k]:|S|\le r-1, 1\in S} \prod_{i\in S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \ge 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \le 0\} \\
&= \sum_{S\subseteq[k]:|S|\le r-1, 1\in S} \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \ge 0\} \cdot \prod_{i\in S\setminus\{1\}} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \ge 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \le 0\} \\
&= \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \ge 0\} \cdot \sum_{S\subseteq[k]:|S|\le r-1, 1\in S} \prod_{i\in S\setminus\{1\}} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \ge 0\} \prod_{i\notin S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \le 0\} \\
&=: \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \ge 0\} \cdot G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\,,
\end{aligned}$$

and similarly

$$\begin{aligned}
T^-(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) &= \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \le 0\} \cdot \sum_{S\subseteq[k]:|S|\le r-1, 1\notin S} \prod_{i\in S} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \ge 0\} \prod_{i\notin S\setminus\{1\}} \mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \le 0\} \\
&=: \mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \le 0\} \cdot G^-(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\,.
\end{aligned}$$

Notice that the indicator G^s does not depend on the alternative 1 for $s \in \{-, +\}$. Since $T : \mathbb{R}^k \to \{0, 1\}$, we have that

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \ne T(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[|T(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) - T(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})|] \\
&\le \sum_{s\in\{-,+\}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[|T^s(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) - T^s(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})|] \\
&= \sum_{s\in\{-,+\}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}[G^s(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \cdot |\mathbf{1}\{s \cdot \boldsymbol{v}_1 \cdot \boldsymbol{x} \ge 0\} - \mathbf{1}\{s \cdot \boldsymbol{r} \cdot \boldsymbol{x} \ge 0\}|]\,.
\end{aligned}$$

Let us focus on the case $s = +$. The difference between the two indicators in the last line of the above equation corresponds to the event that the halfspaces $\mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \geq 0\}$ and $\mathbf{1}\{\boldsymbol{r} \cdot \boldsymbol{x} \geq 0\}$ disagree. Hence, we have that $|\mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} \geq 0\} - \mathbf{1}\{\boldsymbol{r} \cdot \boldsymbol{x} \geq 0\}| = \mathbf{1}\{(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0\}$. Note that the above indicator depends on both $\boldsymbol{v}_1$ and $\boldsymbol{r}$. We would like to work only with one of these two vectors. To this end, let us introduce $\boldsymbol{q}$, the normalized projection of $\boldsymbol{r}$ onto the orthogonal complement of $\boldsymbol{v}_1$, i.e., $\boldsymbol{q} = \mathrm{proj}_{\boldsymbol{v}_1^\perp}\boldsymbol{r}/\|\mathrm{proj}_{\boldsymbol{v}_1^\perp}\boldsymbol{r}\|_2$. We have that $\boldsymbol{v}_1$ and $\boldsymbol{q}$ is an orthonormal basis of the subspace spanned by the vectors $\boldsymbol{v}_1$ and $\boldsymbol{r}$. Notice that $\boldsymbol{r} = \cos\theta(\boldsymbol{v}_1, \boldsymbol{r})\boldsymbol{v}_1 + \sin\theta(\boldsymbol{v}_1, \boldsymbol{r})\boldsymbol{q}$, by the construction of $\boldsymbol{q}$. Our goal is to understand the structure of the region $(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0$. This set is equal to

$$\{0 < \boldsymbol{v}_1 \cdot \boldsymbol{x} < -(\boldsymbol{q} \cdot \boldsymbol{x})\tan\theta(\boldsymbol{v}_1, \boldsymbol{r})\} \cup \{-(\boldsymbol{q} \cdot \boldsymbol{x})\tan\theta(\boldsymbol{v}_1, \boldsymbol{r}) < \boldsymbol{v}_1 \cdot \boldsymbol{x} < 0\}\,.$$

To see this, we have that $(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) = (\boldsymbol{v}_1 \cdot \boldsymbol{x})(\cos\theta(\boldsymbol{v}_1, \boldsymbol{r})\boldsymbol{v}_1 \cdot \boldsymbol{x} + \sin\theta(\boldsymbol{v}_1, \boldsymbol{r})\boldsymbol{q} \cdot \boldsymbol{x})$. This quantity must be negative. The left-hand set considers the case where $\boldsymbol{v}_1 \cdot \boldsymbol{x} > 0$ and so $\tan\theta(\boldsymbol{v}_1, \boldsymbol{r})(\boldsymbol{q} \cdot \boldsymbol{x}) < -\boldsymbol{v}_1 \cdot \boldsymbol{x}$. We obtain the right-hand set in a similar way. Thus, we have that the disagreement region $(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0$ is a subset of the region $\{|\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq |\boldsymbol{q} \cdot \boldsymbol{x}| \tan\theta(\boldsymbol{v}_1, \boldsymbol{r})\}$. Since $\tan\theta(\boldsymbol{v}_1, \boldsymbol{r}) \leq \theta$ and we have that θ is sufficiently small we can also replace the above region by the larger region: $\{|\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq 2\theta|\boldsymbol{q} \cdot \boldsymbol{x}|\}$. Therefore, we have

$$\begin{aligned}&\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\ \mathbf{1}\{(\boldsymbol{v}_1 \cdot \boldsymbol{x})(\boldsymbol{r} \cdot \boldsymbol{x}) < 0\}\right] \\ &\quad\leq \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})\ \mathbf{1}\{|\boldsymbol{v}_1 \cdot \boldsymbol{x}| \leq 2\theta|\boldsymbol{q} \cdot \boldsymbol{x}|\}\right]\,.\end{aligned}$$

From this point, the proof goes as in the top-1 case. In total, we will get that

$$\begin{aligned}&\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[T(\boldsymbol{v}_1 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) \neq T(\boldsymbol{r} \cdot \boldsymbol{x}, \boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}_d}\left[(G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}) + G^-(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x}))\ |\boldsymbol{q} \cdot \boldsymbol{x}|\ \delta(|\boldsymbol{v}_1 \cdot \boldsymbol{x}|)\right] \\ &\leq 2\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}|d\mu(\boldsymbol{x}) \\ &\leq 2\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \leq \xi\}d\mu(\boldsymbol{x}) + 2\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \geq \xi\}d\mu(\boldsymbol{x}) \\ &\leq 2\xi\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})d\mu(\boldsymbol{x}) + 2\int_{\boldsymbol{x}\in F_1}\phi_d(\boldsymbol{x})|\boldsymbol{q} \cdot \boldsymbol{x}|\mathbf{1}\{|\boldsymbol{q} \cdot \boldsymbol{x}| \geq \xi\}d\mu(\boldsymbol{x})\,,\end{aligned}$$

where $d\mu(\boldsymbol{x})$ is the standard surface measure in $\mathbb{R}^d$. Let us explain the first inequality above. Note that the space induced by $G^-(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$ contains the space induced by $G^+(\boldsymbol{v}_2 \cdot \boldsymbol{x}, \ldots, \boldsymbol{v}_k \cdot \boldsymbol{x})$. Hence, in the integration, we can integrate over the surface $F_1 = T(\boldsymbol{V}\boldsymbol{x}) \cap \mathbf{1}\{\boldsymbol{x} : \boldsymbol{v}_1 \cdot \boldsymbol{x} = 0\}$ twice. Essentially, this surface corresponds to $\mathbf{1}\{\boldsymbol{v}_1 \cdot \boldsymbol{x} = 0\} \cdot \sum_{S\subseteq[k]\setminus\{1\}:|S|\leq r-1}\prod_{i\in S}\mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \geq 0\}\prod_{i\notin S}\mathbf{1}\{\boldsymbol{v}_i \cdot \boldsymbol{x} \leq 0\}$. Applying the steps of the top-1 case, we can obtain the desired bound in terms of the Gaussian surface area of F_1. □

Next, for fixed $j \in [k]$, we can apply the above claim sequentially (as we did in the end of the top-1 case) to get

$$\lim_{\theta \to 0} \frac{\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{N}_d}[T_j(\boldsymbol{U}\boldsymbol{x}) \neq T_j(\boldsymbol{V}\boldsymbol{x})]}{\theta} \leq c \cdot \sum_{i \in [k]} \Gamma(F_i^j) \sqrt{\log\left(\frac{1}{\Gamma(F_i^j)} + 1\right)},$$

for some small constant $c > 0$.

5.8.2 The proof of Lemma 5.8.4

Using the above result, we get that it suffices to control the value $\Gamma(F_i^j)$, where F_i^j is the surface of $T_j(\boldsymbol{V}\boldsymbol{x}) \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\}$ for the matrix $\boldsymbol{V}$ and $i, j \in [k]$. We next have to control the Gaussian surface area of the induced shape, i.e., the quantity

$$\Gamma(\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})\} \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\}).$$

To this end, we give the next lemma.

Lemma 5.8.5. *Let $r \leq k$ with $r, k \in \mathbb{N}$. For any matrix $\boldsymbol{V} \in \mathbb{R}^{k \times d}$ and $i, j \in [k]$, there exists a matrix $\boldsymbol{Q} = \boldsymbol{Q}^{(i)} \in \mathbb{R}^{k \times d}$ which depends only on i such that*

$$\Gamma(F_i^j) := \Gamma(\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})\} \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\}) \leq 2 \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})].$$

Before proving this result, let us see how to apply it in order to get Lemma 5.8.4. We will have that

$$\begin{aligned}
\sum_{i \in [k]} \sum_{j \in [k]} \Gamma(F_i^j) &= \sum_{i \in [k]} \sum_{j \in [k]} \Gamma(\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x})\} \cap \{\boldsymbol{x} : \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\}) \\
&\leq 2 \sum_{i \in [k]} \sum_{j \in [k]} \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Q}^{(i)}\boldsymbol{x})] \\
&= 2 \sum_{i \in [k]} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{N}_d}[|\sigma_{1..r}(\boldsymbol{Q}^{(i)}\boldsymbol{x})|] \\
&= 2 \sum_{i \in [k]} r \\
&= 2kr.
\end{aligned}$$

Proof of Lemma 5.8.5. For this proof, we fix $i, j \in [k]$. The first step is to design the matrix $\boldsymbol{Q}$. As a first observation, we can subtract the vector $\boldsymbol{V}_i$ from each weight vector and do not affect the resulting orderings. Second, we can assume that the weight vectors that correspond to indices which j beats are unit. Let us be more specific Assume that initially we have that

$$(\boldsymbol{V}_j - \boldsymbol{V}_\ell) \cdot \boldsymbol{x} \geq 0.$$

The first observation gives that

$$(\boldsymbol{V}_j - \boldsymbol{V}_i) \cdot \boldsymbol{x} \geq (\boldsymbol{V}_\ell - \boldsymbol{V}_i) \cdot \boldsymbol{x} \,.$$

Let us set $\widetilde{\boldsymbol{Q}}$ the intermediate matrix with rows $\boldsymbol{V}_j - \boldsymbol{V}_i$. The second observation states that the inequalities where j beats some index ℓ are not affected by normalization. Note that $\widetilde{\boldsymbol{Q}}_j \cdot \boldsymbol{x} = 0$ and hence $\widetilde{\boldsymbol{Q}}_\ell \cdot \boldsymbol{x} \leq 0$. Hence, dividing with non-negative numbers will not affect the order of these two values, i.e.,

$$\frac{\widetilde{\boldsymbol{Q}}_j \cdot \boldsymbol{x}}{\|\widetilde{\boldsymbol{Q}}_j\|_2} \geq \frac{\widetilde{\boldsymbol{Q}}_\ell \cdot \boldsymbol{x}}{\|\widetilde{\boldsymbol{Q}}_\ell\|_2} \,.$$

Note that the above ordering is $\boldsymbol{x}$-dependent, since the indices that j beats depend on $\boldsymbol{x}$. However, we can normalize any row of $\widetilde{\boldsymbol{Q}}$ without affecting the fact that the element j is top-r (since the sign of the inner products is not affected by normalization). This transformation yields a matrix $\boldsymbol{Q} = \boldsymbol{Q}^{(i)}$ and depends only on i (crucially, it is independent of j). For simplicity, we will omit the index i in what follows. For this matrix, we have that

$$\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x}), \boldsymbol{Q}_j \cdot \boldsymbol{x} = 0\} = \{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{V}\boldsymbol{x}), \boldsymbol{V}_i \cdot \boldsymbol{x} = \boldsymbol{V}_j \cdot \boldsymbol{x}\} \,.$$

We will now prove that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d} [j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})] \geq \frac{\Gamma(F_i^j)}{2} \,.$$

Let us fix some $\boldsymbol{x}$ and set $\boldsymbol{x}^{\|} = \mathrm{proj}_{\boldsymbol{Q}_j} \boldsymbol{x}$ and $x^{\perp} = \mathrm{proj}_{\boldsymbol{Q}_j^{\perp}} \boldsymbol{x}$. We assume that $\boldsymbol{x}$ lies in the set $\{\boldsymbol{x} : j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})\}$. This implies that there exist an index set I of size at least $k - r$ so that if $\ell \in I$ then

$$\boldsymbol{Q}_j \cdot \boldsymbol{x}^{\|} + \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\perp} \geq \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\|} + \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\perp} \,.$$

Let us condition on the event

$$\boldsymbol{Q}_j \cdot \boldsymbol{x}^{\perp} \geq \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\perp} \,.$$

We hence get that

$$\boldsymbol{Q}_j \cdot \boldsymbol{x}^{\|} = (\boldsymbol{Q}_j \cdot \boldsymbol{Q}_j) \cdot (\boldsymbol{Q}_j \cdot \boldsymbol{x}) \geq \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\|} = (\boldsymbol{Q}_\ell \cdot \boldsymbol{Q}_j) \cdot (\boldsymbol{Q}_j \cdot \boldsymbol{x})$$

Using that $\boldsymbol{Q}_j$ is unit, that the inner product between $\boldsymbol{Q}_\ell$ and $\boldsymbol{Q}_j$ is at most one and that $\boldsymbol{Q}_j \cdot \boldsymbol{x}$ is a univariate Gaussian, we get that

$$\mathop{\mathbf{Pr}}_{z \sim \mathcal{N}(0,1)} [z \cdot (1 - \boldsymbol{Q}_\ell \cdot \boldsymbol{Q}_j) \geq 0] = 1/2 \,.$$

The above discussion implies that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d} [j \in \sigma_{1..r}(\boldsymbol{Q}\boldsymbol{x})] = \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{N}_d} [(\forall \ell \in I)\ \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\|} + \boldsymbol{Q}_j \cdot \boldsymbol{x}^{\perp} \geq \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\|} + \boldsymbol{Q}_\ell \cdot \boldsymbol{x}^{\perp}]$$

and so $\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Qx})]$ equals to

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[(\forall \ell \in I)\, \boldsymbol{Q}_j{\cdot}\boldsymbol{x}^{\|} \geq \boldsymbol{Q}_j{\cdot}\boldsymbol{x}^{\|} \mid (\forall \ell \in I)\, \boldsymbol{Q}_j{\cdot}\boldsymbol{x}^{\perp} \geq \boldsymbol{Q}_\ell{\cdot}\boldsymbol{x}^{\perp}]{\cdot} \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[(\forall \ell \in I)\, \boldsymbol{Q}_j{\cdot}\boldsymbol{x}^{\perp} \geq \boldsymbol{Q}_\ell{\cdot}\boldsymbol{x}^{\perp}]\,.$$

However, in the above product, we have that the first term is 1/2 and the second term is the probability that $j \in \sigma_{1..r}(\boldsymbol{Qx}^{\perp})$, i.e.,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}_d}[j \in \sigma_{1..r}(\boldsymbol{Qx})] \geq \frac{\mathbf{Pr}[j \in \sigma_{1..r}(\boldsymbol{Qx}^{\perp})]}{2} = \Gamma(F_i^j)/2\,,$$

since the space in the RHS is low-dimensional and corresponds to the desired surface. □

5.9 Distribution-Free Lower Bounds for Top-1 Disagreement Error

We begin with some definitions concerning the PAC Label Ranking setting. Let $\mathcal{X}$ be an instance space and $\mathcal{Y} = \mathbb{S}_k$ be the space of labels, which are rankings over k elements. A sorting function or hypothesis is a mapping $h : \mathcal{X} \to \mathbb{S}_k$. We denote by $h_1(x)$ the top-1 element of the ranking $h(x)$. A hypothesis class is a set of classifiers $\mathcal{H} \subset \mathbb{S}_k^{\mathcal{X}}$.

Top-1 Disagreement Error. The top-1 disagreement error with respect to a joint distribution $\mathcal{D}$ over $\mathcal{X} \times \mathbb{S}_k$ equals to the probability $\mathbf{Pr}_{(x,\sigma)\sim\mathcal{D}}[h_1(x) \neq \sigma^{-1}(1)]$. We mainly consider learning in the **realizable** case, which means that there is $h^\star \in \mathcal{H}$ which has (almost surely) zero error. Therefore, we can focus on the marginal distribution $\mathcal{D}_x$ over $\mathcal{X}$ and denote the top-1 disagreement error of a sorting function h with respect to the true hypothesis $h^\star$ by $\mathrm{Err}_{\mathcal{D}_x,h^\star}(h) := \mathbf{Pr}_{x\sim\mathcal{D}_x}[h_1(x) \neq h_1^\star(x)]$.

A learning algorithm is a function $\mathcal{A}$ that receives a training set of m instances, $S \in \mathcal{X}^m$, together with their labels according to $h^\star$. We denote the restriction of $h^\star$ to the instances in S by $h^\star|_S$. The output of the algorithm $\mathcal{A}$, denoted $\mathcal{A}(S, h^\star|_S)$ is a sorting function. A learning algorithm is proper if it always outputs a hypothesis from $\mathcal{H}$.

The top-1 PAC Label Ranking sample complexity of a learning algorithm $\mathcal{A}$ is the function $m^{(1)}_{\mathcal{A},\mathcal{H}}$ defined as follows: for every $\epsilon, \delta > 0$, $m^{(1)}_{\mathcal{A},\mathcal{H}}(\epsilon, \delta)$ is the minimal integer such that for every $m \geq m^{(1)}_{\mathcal{A},\mathcal{H}}(\epsilon, \delta)$, every distribution $\mathcal{D}_x$ on $\mathcal{X}$, and every target hypothesis $h^\star \in \mathcal{H}$, $\mathbf{Pr}_{S\sim\mathcal{D}_x^m}[\mathrm{Err}_{\mathcal{D}_x,h^\star}(\mathcal{A}(S, h^\star|_S)) > \epsilon] \leq \delta$. In this case, we say that the learning algorithm (ϵ, δ)-learns the class of sorting functions $\mathcal{H}$ with respect to the top-1 disagreement error. If no integer satisfies the inequality above, define $m^{(1)}_{\mathcal{A}}(\epsilon, \delta) = \infty$. $\mathcal{H}$ is learnable with $\mathcal{A}$ if for all ϵ and δ the sample complexity is finite. The **top-1 PAC Label Ranking sample complexity** of a class $\mathcal{H}$ is $m^{(1)}_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta) = \inf_{\mathcal{A}} m^{(1)}_{\mathcal{A},\mathcal{H}}(\epsilon, \delta)$, where the infimum is taken over all learning algorithms. Clearly, the above top-1 definition can be extended to the top-r setting.

In this section, we show the next result. We denote by $\mathcal{L}_{d,k}$ the class of Linear Sorting functions in d dimensions with k labels.

Theorem 5.9.1. *In the realizable PAC Label Ranking setting, any algorithm that (ϵ, δ)-learns the class $\mathcal{L}_{d,k}$ with respect to the top-1 disagreement error requires at least $\Omega((dk + \log(1/\delta))/\epsilon)$ samples.*

5.9.1 Top-1 Ranking Natarajan Dimension

In order to establish the above result, we introduce a variant of the standard Natarajan dimension (Nat89; BDCBL92; DSBDSS11; DSS14). For a ranking π, we will also let $L_1(\pi)$ its top-1 element and $L_{3..k}(\pi)$ the ranking after deleting its top-2 part.

Definition 5.9.2 (Top-1 Ranking Natarajan Dimension)**.** *Let $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$ be a hypothesis class of sorting functions and let $S \subseteq \mathcal{X}$. We say that $\mathcal{H}$ N-shatters S if there exist two mappings $f_1, f_2 : S \to \mathbb{S}_k$ such that for every $y \in S$, $L_1(f_1(y)) \neq L_1(f_2(y))$ and $L_{3..k}(f_1(y)) = L_{3..k}(f_2(y))$ and for every $T \subseteq S$, there exists a sorting function $g \in \mathcal{H}$ such that*

$$(i)\ \forall x \in T, \quad g(x) = f_1(x), \text{ and } (ii)\ \forall x \in S \setminus T, \quad g(x) = f_2(x)\,.$$

The ***top-1 Ranking Natarajan dimension*** *of $\mathcal{H}$, denoted $d_N^{(1)}(\mathcal{H})$ is the maximal cardinality of a set that is N-shattered by $\mathcal{H}$.*

First, we connect PAC Label Ranking learnability to the top-1 disagreement error with the notion of top-1 Ranking Natarajan dimension.

Theorem 5.9.3 (Top-1-Natarajan Lower Bounds Sample Complexity)**.** *In the realizable PAC Label Ranking setting, we have for every hypothesis class $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$*

$$m^{(1)}_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta) = \Omega\left(\frac{d_N^{(1)}(\mathcal{H}) + \log(1/\delta)}{\epsilon}\right)\,.$$

Proof. Let $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$ be a class of sorting functions of top-1-Natarajan dimension $d_N^{(1)} = d_N$. Consider the binary hypothesis class $\mathcal{H}_{\mathrm{bin}} = \{0,1\}^{[d_N]}$ which contains all the classifiers from $[d_N] = \{1, ..., d_N\}$ to $\{0,1\}$. It suffices to show the following.

Claim 18. *It holds that $m^{(1)}_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta) \geq m_{\mathrm{PAC},\mathcal{H}_{\mathrm{bin}}}(\epsilon, \delta)$.*

This is sufficient since we have that $m_{\mathrm{PAC},\mathcal{H}_{\mathrm{bin}}}(\epsilon, \delta) = \Omega\left(\frac{\mathrm{VC}(\mathcal{H}_{\mathrm{bin}}) + \log(1/\delta)}{\epsilon}\right)$ and $\mathrm{VC}(\mathcal{H}_{\mathrm{bin}}) = d_N$. Let us now prove the claim.

We assume that the instance space is the set $\mathcal{X}$. Assume that A is a learning algorithm for the hypothesis class $\mathcal{H} \subseteq \mathbb{S}_k^{\mathcal{X}}$ and A_{bin} is a learning algorithm for the associated binary class $\mathcal{H}_{\text{bin}}$. It suffices to show that A requires at least as many samples as A_{bin}. In fact, we will show that whenever A_{bin} errs, so does A. Let $S = \{s_1, ..., s_{d_N}\}, f_0, f_1$ be the set and the two functions that witness that the top-1-Natarajan dimension of $\mathcal{H}$ is d_N. Given a training set $(x_i, y_i)_{i\in[m]} \in ([d_N] \times \{0,1\})^m$, we set $g : \mathcal{X} \to \mathbb{S}_k$ be equal to the output of the algorithm A with input $(s_{x_i}, f_{y_i}(x_i))_{i\in[m]} \in (S \times \mathbb{S}_k)^m$. We also set f be the output of the algorithm A_{bin} with input $(x_i, y_i)_{i\in[m]}$ by setting $f(i) = 1$ if and only if $L_1(g(s_i)) = L_1(f_1(s_i))$. We will show that whenever A_{bin} errs, so does A. Fix $(x_i, y_i) \in S \times \{0,1\}$. Assume that $A_{\text{bin}}(x_i) \neq y_i$ and say $y_i = 0$. Then $f(i) = 1$ and so $L_1(g(s_i)) = L_1(f_1(s_i)) \neq L_1(f_0(s_i))$. This implies that A errs. The case $y_i = 1$ is similar. □

5.9.2 Lower Bound for top-1 disagreement error for LSFs

Theorem 5.9.4 (Top-1 Natarajan Dimension of LSFs). *Consider the hypothesis class $\mathcal{L}_{d,k} = \{\sigma_{\boldsymbol{W}} : \mathbb{R}^d \to \mathbb{S}_k : \sigma_{\boldsymbol{W}}(\boldsymbol{x}) = \text{argsort}(\boldsymbol{W}\boldsymbol{x}), \boldsymbol{W} \in \mathbb{R}^{k\times d}\}$. Then, $d_N^{(1)}(\mathcal{L}_{d,k}) = \Omega(dk)$.*

Proof. Fix $k \in \mathbb{N}$. Let us consider the case $d = 2$ that will correspond as the building block for the general case $d > 2$. Let us first choose the set of points: Set P be the collection of pairs $P = \{(2i-1, 2i)\}_{i\in[b]}$ for any $i \in [b]$ with $b = \lfloor k/2 \rfloor$ and $S = \{\boldsymbol{x}_m\}_{m\in P}$ where these points correspond to $|P|$ equidistributed points on the unit sphere in $\mathbb{R}^2$. This set of points has size $|P| = \Theta(k)$ and we are going to N-shatter it using $\mathcal{L}_{2,k}$.

Consider the matrix $\boldsymbol{W} \in \mathbb{R}^{k\times 2}$ so that $\{\boldsymbol{W}_i\}_{i\in[k]}$ correspond to the rows of $\boldsymbol{W}$. The structure of the problem relies on the hyperplanes with normal vectors $(\boldsymbol{W}_i - \boldsymbol{W}_j)_{i\neq j}$ and our choice of $\boldsymbol{W}$ will rely on these hyperplanes. For any $m = (2i-1, 2i)$, we set $\boldsymbol{W}_{2i-1}, \boldsymbol{W}_{2i}$ on the unit sphere so that $\boldsymbol{W}_{2i-1} \cdot \boldsymbol{W}_{2i} = 1 - \phi$ with $\phi \in (0,1)$ sufficiently small (set $\arccos(1-\phi) = 2\pi/(100k)$) and let C_m be the cone generated by these two vectors with axis I_m. We place $\boldsymbol{W}_{2i-1}$ so that the distance between $\boldsymbol{x}_m$ and the hyperplane I_m is sufficiently small (say that the angle between $\boldsymbol{x}_m$ and I_m is $\arccos(1-\phi)/100$). Note that the normal vector of I_m is $\boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i}$ and we place $\boldsymbol{x}_m$ so that it has positive correlation with this vector. This uniquely identifies the location of $\boldsymbol{W}_{2i}$. Crucially, each vector $\boldsymbol{x}_m$ has the following properties: (i) $\boldsymbol{x}_m$ is very close to the boundary of the hyperplane with normal vector $(\boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i})$, (ii) $\boldsymbol{W}_{2i-1} \cdot \boldsymbol{x}_m > \boldsymbol{W}_{2i} \cdot \boldsymbol{x} > \boldsymbol{W}_j \cdot \boldsymbol{x}_m$ for any $j \notin m$ and (iii) $\boldsymbol{x}_m$ is far from any boundary induced by hyperplanes with normal vectors $\boldsymbol{W}_j - \boldsymbol{W}_{j'}$ for any $(j, j') \neq m$.

Since the points are well-separated on the unit sphere, for any $m = (2i-1, 2i) \in P$, we have $\boldsymbol{W}_{2i-1} \cdot \boldsymbol{W}_{2i} = 1 - \phi \approx 1$ and for any other pair of indices $(i,j) \notin P$, there exists $c = c(k) \in (0,1)$, $|\langle \boldsymbol{W}_i, \boldsymbol{W}_j \rangle| \leq c$.

For any $m = (2i-1, 2i) \in P$, we set $\boldsymbol{W}'_{2i-1} - \boldsymbol{W}'_{2i} = \boldsymbol{R}_\theta(\boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i})$ for some θ to be chosen, where $\boldsymbol{R}_\theta$ is the 2×2 rotation matrix. We choose θ so that each point $\boldsymbol{x}_m$ for $m = (2i-1, 2i) \in P$ with $(\boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i}) \cdot \boldsymbol{x}_m > 0$ satisfies $(\boldsymbol{W}'_{2i-1} - \boldsymbol{W}'_{2i}) \cdot \boldsymbol{x}_m < 0$. The main idea is that since $\boldsymbol{x}_m$ has the properties (i)-(iii) described above, the rankings induced by the vectors $\boldsymbol{W}\boldsymbol{x}_m$ and $\boldsymbol{W}'\boldsymbol{x}_m$ will be different in the first two positions but the same in the rest.

Given the training set $\{\boldsymbol{x}_m\}_{m \in P}$, we have to construct f_0, f_1 and verify that they satisfy the top-1 Ranking Natarajan conditions. For $m = (2i-1, 2i)$, we have that $f_0(\boldsymbol{x}_m) = (2i-1, 2i, \pi)$ and $f_1(\boldsymbol{x}_m) = (2i, 2i-1, \pi)$ for some ranking π of size $k-2$ that depends on m. Specifically, we will set $f_0(\boldsymbol{x}) = \sigma(\boldsymbol{W}\boldsymbol{x})$ and $f_1(\boldsymbol{x}) = \sigma(\boldsymbol{W}'\boldsymbol{x})$, where σ gives the decreasing ordering of the elements of the input vector. By the choice of the set S and $\boldsymbol{W}, \boldsymbol{W}'$, it remains to show that the $k-2$ last elements of the rankings $f_0(\boldsymbol{x}_m)$ (say π_0) and of $f_1(\boldsymbol{x}_m)$ (say π_1) are in the same order, i.e., $L_{3..k}(f_0(\boldsymbol{x}_m)) = L_{3..k}(f_1(\boldsymbol{x}_m))$. Assume that $u \succ v$ in π_0. It suffices to show that $(\boldsymbol{W}'_u - \boldsymbol{W}'_v) \cdot \boldsymbol{x}_m \geq 0$, i.e., the order of u and v is preserved when transforming $\boldsymbol{W}$ to $\boldsymbol{W}'$. We have that $(\boldsymbol{W}_u - \boldsymbol{W}_v) \cdot \boldsymbol{x}_m > c_1$ for some constant $c_1 > 0$ (c_1 is the minimum over $(u, v) \neq m = (2i-1, 2i)$). Hence, we can pick θ small enough so that $(\boldsymbol{W}'_u - \boldsymbol{W}'_v) \cdot \boldsymbol{x}_m > c_2$ and this can be done for any pair u, v that does not correspond to m. This implies that $\pi_0 = \pi_1 = \pi$. In particular, we have that

$$(\boldsymbol{W}'_u - \boldsymbol{W}'_v) \cdot \boldsymbol{x}_m = \cos(\theta) \cdot (\boldsymbol{W}_u - \boldsymbol{W}_v) \cdot \boldsymbol{x}_m + \sin(\theta) \cdot (W^{(1)}_{uv} x^{(2)}_m - W^{(2)}_{uv} x^{(1)}_m) > c_2 > 0$$

for some θ sufficiently small, where $W^{(t)}_{uv}$ is the t-th entry of the vector $\boldsymbol{W}_u - \boldsymbol{W}_v$ for $t \in \{1, 2\}$ and $\boldsymbol{x}_m, \boldsymbol{W}_u, \boldsymbol{W}_v$ are unit vectors.

For any subset T of S, it remains to choose a linear classifier in $\mathcal{L}_{2,k}$ (which is allowed to depend on T). For any $T \subseteq S = \{\boldsymbol{x}_m\}_{m \in P}$, we consider the matrix $\overline{\boldsymbol{W}} \in \mathbb{R}^{k \times 2}$ so that for the i-th row $\overline{W}_i = \boldsymbol{W}_i 1\{i \in m \in T\} + \boldsymbol{W}'_i 1\{i \in m \in S \setminus T\}$ for any $i \in [k]$. This is valid since the pairs $m \in P$ partition $[k]$. We have to show the following two properties: (i) $\sigma(\overline{\boldsymbol{W}}\boldsymbol{x}) = f_0(\boldsymbol{x})$ for $x \in T$ and (ii) $\sigma(\overline{\boldsymbol{W}}\boldsymbol{x}) = f_1(\boldsymbol{x})$ for $x \in S \setminus T$.

Assume that $m = (2i-1, 2i)$ and $\boldsymbol{x}_m \in T$. We have that $f_0(\boldsymbol{x}_m) = (2i-1, 2i, \pi)$ and $\overline{\boldsymbol{W}}_{2i-1} - \overline{\boldsymbol{W}}_{2i} = \boldsymbol{W}_{2i-1} - \boldsymbol{W}_{2i}$ and so $2i-1 \succ 2i$ in the ranking $\sigma(\overline{\boldsymbol{W}}\boldsymbol{x}_m)$. It remains to show that the remaining $\binom{k}{2} - 1$ pairwise comparisons are the same in the two rankings. Let us consider a pair of points $u \neq v$ so that $u \succ v$ in $f_0(\boldsymbol{x}_m)$. It suffices to show that $u \succ v$ in $\sigma(\overline{\boldsymbol{W}}\boldsymbol{x}_m)$.

1. If u, v are so that $\overline{\boldsymbol{W}}_u - \overline{\boldsymbol{W}}_v = \boldsymbol{W}_u - \boldsymbol{W}_v$, the result holds.

2. If u, v are so that $\overline{\boldsymbol{W}}_u - \overline{\boldsymbol{W}}_v = \boldsymbol{W}_u - \boldsymbol{W}'_v$: In this case, u and v lie in a different pair of P and this implies that the correct direction is preserved if θ is appropriately chosen. For θ as above, it holds that $(\boldsymbol{W}_u - \boldsymbol{R}_\theta \boldsymbol{W}_v) \cdot \boldsymbol{x}_m$ has the same sign as $(\boldsymbol{W}_u - \boldsymbol{W}_v) \cdot \boldsymbol{x}_m$. In particular,

   $$\boldsymbol{W}_u \cdot \boldsymbol{x}_m - \boldsymbol{R}_\theta \boldsymbol{W}_v \cdot \boldsymbol{x}_m = \boldsymbol{W}_u \cdot \boldsymbol{x}_m - (\cos(\theta) W^{(1)}_v - \sin(\theta) W^{(2)}_v) x^{(1)}_m - (\sin(\theta) W^{(1)}_v + \cos(\theta) W^{(2)}_v) x^{(2)}_m \,,$$

   and so

   $$(\boldsymbol{W}_u - \boldsymbol{W}'_v) \cdot \boldsymbol{x}_m = \cos(\theta) \cdot (\boldsymbol{W}_u - \boldsymbol{W}_v) \cdot \boldsymbol{x}_m + \sin(\theta)(W^{(2)}_v x^{(1)}_m - W^{(1)}_v x^{(2)}_m) > 0 \,.$$

3. If u, v are so that $\overline{\boldsymbol{W}}_u - \overline{\boldsymbol{W}}_v = \boldsymbol{W}'_u - \boldsymbol{W}'_v$, the analysis for the inner product with $\boldsymbol{x}_m$ will be similar.

We now have to extend this proof for $d > 2$. We will "tensorize" the above construction as follows. Let $S = \{\boldsymbol{y}_{mj}\}_{m \in [b], j \in [d/2]}$ with $|S| = \lfloor k/2 \rfloor \cdot \lfloor d/2 \rfloor$. We first define the points of S: For $s \in [d]$, set $y_{mj}[s] = x_m[1] 1\{s = 2j-1\} + x_m[2] 1\{s = 2j\}$ with $\boldsymbol{y}_{mj} \in \mathbb{R}^d$, i.e., $\boldsymbol{y}_{mj}$ has the values of $\boldsymbol{x}_m$ at the consecutive entries indicated by $m = (2i-1, 2i) \in P$ and zeros at the other positions.

We have to show that the set S is N-shattered. Given $T \subseteq S$, we are going to create the matrix $\overline{\boldsymbol{W}} \in \mathbb{R}^{k \times d}$. For illustration, think of each row of the matrix as having $d/2$ blocks of size two. If $\boldsymbol{y}_{mj} \in T$ with $m = (2i-1, 2i)$, set the two associated rows (indicated by m) of $\overline{\boldsymbol{W}}$ with $\boldsymbol{W}_{2i-1}, \boldsymbol{W}_{2i}$ at the j-th block and with $\boldsymbol{W}'_{2i-1}, \boldsymbol{W}'_{2i}$ otherwise. We will have that $\sigma(\overline{\boldsymbol{W}}\boldsymbol{y}) = f_0(\boldsymbol{y})$ if $y \in T$ and $\sigma(\overline{\boldsymbol{W}}\boldsymbol{y}) = f_1(\boldsymbol{y})$ otherwise and the analysis is the same as the $d = 2$ case. □

5.10 Examples of Noisy Ranking Distributions

Definition 5.10.1 (Mallows model (Mal57))**.** *Consider k alternatives and let $\pi \in \mathbb{S}_k, \phi \in [0,1]$. The Mallows distribution $\mathcal{M}_{\text{Mal}}(\pi, \phi)$ with central ranking π and spread parameter ϕ is a probability measure over $\mathbb{S}_k$ with density $\mathbf{Pr}_{\sigma \sim \mathcal{M}_{\text{Mal}}(\pi,\phi)}[\sigma]$ that is proportional to $\phi^{d(\sigma,\pi)}$, where d is a ranking distance.*

We focus on Mallows models accociated with the Kendall's Tau distance $d = d_{KT}$ (the standard distance, not the normalized one), which measures the number of discordant pairs.

Fact 5. *When $\phi < 1$, the Mallows model $\mathcal{M}_{\text{Mal}}(\pi, \phi)$ is a ranking distribution with bounded noise at most $\frac{1+\phi}{4} < 1/2$.*

Proof. The following property holds (Mal57)

$$\mathop{\mathbf{Pr}}_{\sigma \sim \mathcal{M}_{\text{Mal}}(\pi,\phi)}[\sigma(i) < \sigma(j) | \pi(i) < \pi(j)] = \frac{\pi(j) - \pi(i) + 1}{1 - \phi^{\pi(j)-\pi(i)+1}} - \frac{\pi(j) - \pi(i)}{1 - \phi^{\pi(j)-\pi(i)}} \geq \frac{1}{2} + \frac{1-\phi}{4}\,.$$

□

The Bradley-Terry-Luce model (BT52b; Luc12) is the most studied pairwise comparisons model. In his seminal paper, Mallows (Mal57) also studied the following natural ranking distribution:

Definition 5.10.2 (Bradley-Terry-Mallows (Mal57))**.** *Consider a score vector $\boldsymbol{w} \in \mathbb{R}^k_+$ with k distinct entries and let π be the ranking induced by the values of $\boldsymbol{w}$ in decreasing order. The Bradley-Terry-Mallows distribution $\mathcal{M}_{\text{BTM}}(\boldsymbol{w})$ with central ranking π is a probability measure over $\mathbb{S}_k$ with density $\mathbf{Pr}_{\sigma \sim \mathcal{M}_{\text{BTM}}(\boldsymbol{w})}[\sigma]$ that is proportional to $\prod_{i \succ_\sigma j} \frac{w_i}{w_i + w_j}$.*

Lemma 5.10.3. *There exists a real number* $0 < \eta < 1/2$ *so that the Bradley-Terry-Mallows distribution* $\mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})$ *is a ranking distribution with bounded noise at most* η*.*

Proof. In the standard Bradley-Terry-Luce model, the pairwise comparison between the alternatives i, j is a Bernoulli random variable with $\mathbf{Pr}[i \succ j] = w_i/(w_i + w_j)$. The Bradley-Terry-Mallows distribution can be considered as the Bradley-Terry-Luce model conditioned on the event that all the pairwise comparisons are consistent to a ranking. Hence, we have that

$$\underset{\sigma \sim \mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})}{\mathbf{Pr}}[\sigma] = \frac{1}{Z(k, \boldsymbol{w})} \prod_{i \succ_\sigma j} \frac{w_i}{w_i + w_j} .$$

Let us set $\mathcal{A}_{i \succ j} = \{\sigma \in \mathbb{S}_k : \sigma(i) < \sigma(j)\}$. We are interested in the following probability

$$\begin{aligned} \underset{\sigma \sim \mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})}{\mathbf{Pr}}[i \succ_\sigma j | w_i > w_j] &= \underset{\sigma \sim \mathcal{M}_{\mathrm{BTM}}(\boldsymbol{w})}{\mathbf{Pr}}[\sigma(i) < \sigma(j) | w_i > w_j] \\ &= \frac{1}{Z(k, \boldsymbol{w})} \sum_{\sigma \in \mathcal{A}_{i \succ j}} \prod_{p \succ_\sigma q} \frac{w_p}{w_p + w_q} . \end{aligned}$$

Note that in order to show the desired property, it suffices to show that

$$\sum_{\sigma \in \mathcal{A}_{i \succ j}} \prod_{p \succ_\sigma q} \frac{w_p}{w_p + w_q} > \sum_{\sigma \in \mathcal{A}_{i \prec j}} \prod_{p \succ_\sigma q} \frac{w_p}{w_p + w_q} .$$

First, observe that there exists a correspondence mapping $\sigma \in \mathcal{A}_{i \succ j}$ to $\mathcal{A}_{i \prec j}$, where one flips the elements i and j. Hence, it suffices to show that the mass of the ranking $(u_a)i(u_b)j(u_c)$ is larger than the one of the ranking $(u_a)j(u_b)i(u_c)$, where u_a, u_b, u_c are permutations of length between 0 and $k-2$ with elements in $[k] \setminus \{i, j\}$. For the two above rankings, the only terms of the product that are not identical are the following

$$\frac{w_i}{w_i + w_j} \prod_{x \in u_b} \frac{w_i}{w_i + w_x} \frac{w_x}{w_x + w_j} > \frac{w_j}{w_i + w_j} \prod_{x \in u_b} \frac{w_j}{w_j + w_x} \frac{w_x}{w_x + w_i} ,$$

since $w_i > w_j$ and so the result follows. □

Chapter 6

Replicable Bandits

In this chapter, we study the notion of replicability in the context of interactive learning and, in particular, in the fundamental setting of stochastic bandits.

Stochastic multi-armed bandits for the general setting without structure have been studied extensively (Sli19; LS20; BCB$^+$12; ACBF02; CBF98; KCG12; ABM10; AG12; KKM12). In this setting, the optimum regret achievable is $O\left(\log(T) \sum_{i:\Delta_i>0} \Delta^{-1}\right)$; this is achieved, e.g., by the upper confidence bound (UCB) algorithm of (ACBF02). The setting of d-dimensional linear stochastic bandits is also well-explored (DHK08; AYPS11) under the well-specified linear reward model, achieving (near) optimal problem-independent regret of $O(d\sqrt{T\log(T)})$ (LS20). Note that the best-known lower bound is $\Omega(d\sqrt{T})$ (DHK08) and that the number of arms can, in principle, be unbounded. For a finite number of arms K, the best known upper bound is $O(\sqrt{dT\log(K)})$ (BCBK12). In general, there is also extensive work in adversarial bandits and we refer the interested reader to (LS20).

Our work focuses on the design of replicable bandit algorithms and we hence consider only stochastic environments. We now remind to the reader our definition of replicability in the bandit setting.

Definition 6.0.1 (Replicable Bandit Algorithm)**.** *Let $\rho \in [0,1]$. We call a bandit algorithm $\mathbb{A}$ ρ-replicable in the stochastic setting if for any distribution $\mathcal{D}_{a_j}$ over $[0,1]$ of the rewards of the j-th arm $a_j \in \mathcal{A}$, and for any two executions of $\mathbb{A}$, where the internal randomness ξ is shared across the executions, it holds that*

$$\mathop{\mathbf{Pr}}_{\xi, \boldsymbol{r}^{(1)}, \boldsymbol{r}^{(2)}}\left[\left(a_1^{(1)}, \ldots, a_T^{(1)}\right) = \left(a_1^{(2)}, \ldots, a_T^{(2)}\right)\right] \geq 1 - \rho\,.$$

Here, $a_t^{(i)} = \mathbb{A}(a_1^{(i)}, r_1^{(i)}, ..., a_{t-1}^{(i)}, r_{t-1}^{(i)}; \xi)$ is the t-th action taken by the algorithm $\mathbb{A}$ in execution $i \in \{1,2\}$.

Our results are summarized in the next table.

Summary of Results			
Setting	Algorithm	Regret	Theorem
Stochastic MAB	Algorithm 8	$\widetilde{O}\left(\frac{K^2\log^3(T)H_\Delta}{\rho^2}\right)$	Theorem 6.2.1
Stochastic MAB	Algorithm 9	$\widetilde{O}\left(\frac{K^2\log(T)H_\Delta}{\rho^2}\right)$	Theorem 6.3.1
Stochastic Linear Bandits	Algorithm 10	$\widetilde{O}\left(\frac{K^2\sqrt{dT}}{\rho^2}\right)$	Theorem 6.4.2
Stochastic Linear Bandits Infinite Action Space	Algorithm 11	$\widetilde{O}\left(\frac{\mathrm{poly}(d)\sqrt{T}}{\rho^2}\right)$	Theorem 6.4.6

Table 6.1: Our results for replicable stochastic general multi-armed and linear bandits. In the expected regret column, $\widetilde{O}(\cdot)$ subsumes logarithmic factors. H_Δ is equal to $\sum_{j:\Delta_j>0} 1/\Delta_j$, Δ_j is the difference between the mean of action j and the optimal action, K is the number of arms, d is the ambient dimension in the linear bandit setting.

6.1 Stochastic Bandits and Replicability

In this section, we first highlight the main challenges in order to guarantee replicability and then discuss how the results of (ILPS22) can be applied in our setting.

6.1.1 Warm-up I: Naive Replicability and Challenges

Let us consider the stochastic two-arm setting ($K = 2$) and a bandit algorithm $\mathbb{A}$ with two independent executions, $\mathbb{A}_1$ and $\mathbb{A}_2$. The algorithm $\mathbb{A}_i$ plays the sequence $1, 2, 1, 2, \ldots$ until some, potentially random, round $T_i \in \mathbb{N}$ after which one of the two arms is eliminated and, from that point, the algorithm picks the winning arm $j_i \in \{1, 2\}$. The algorithm $\mathbb{A}$ is ρ-replicable if and only if $T_1 = T_2$ and $j_1 = j_2$ with probability $1 - \rho$.

Assume that $|\mu_1 - \mu_2| = \Delta$ where μ_i is the mean of the distribution of the i-th arm. If we assume that Δ is known, then we can run the algorithm for $T_1 = T_2 = \frac{C}{\Delta^2}\log(1/\rho)$ for some universal constant $C > 0$ and obtain that, with probability $1-\rho$, it will hold that $\widehat{\mu}_1^{(j)} \approx \mu_1$ and $\widehat{\mu}_2^{(j)} \approx \mu_2$ for $j \in \{1, 2\}$, where $\widehat{\mu}_i^{(j)}$ is the estimation of arm's i mean during execution j. Hence, knowing Δ implies that the stopping criterion of the algorithm $\mathbb{A}$ is deterministic and that, with high probability, the winning arm will be detected at time $T_1 = T_2$. This will make the algorithm ρ-replicable.

Observe that when $K = 2$, the only obstacle to replicability is that the algorithm should decide at the same time to select the winning arm and the selection must be the same in the two execution threads. In the presence of multiple arms, there exists the additional constraint that the above conditions must be satisfied during, potentially, multiple arm eliminations. Hence, the two questions arising from the above discussion are (i) how to modify the above approach when Δ is unknown and (ii) how to deal with $K > 2$ arms.

A potential solution to the second question (on handling $K > 2$ arms) is the Execute-Then-Commit (ETC) strategy. Consider the stochastic K-arm bandit setting. For any $\rho \in (0,1)$, the ETC algorithm with known $\Delta = \min_i \Delta_i$ and horizon T that uses $m = \frac{4}{\Delta^2}\log(1/\rho)$ deterministic exploration phases before commitment is ρ-replicable. The intuition is exactly the same as in the $K = 2$ case. The caveats of this approach are that it assumes that Δ is known and that the obtained regret is quite unsatisfying. In particular, it achieves regret bounded by $m\sum_{i\in[K]}\Delta_i + \rho \cdot (T - mK)\sum_{i\in[k]}\Delta_i$.

Next, we discuss how to improve the regret bound without knowing the gaps Δ_i. Before designing new algorithms, we will inspect the guarantees that can be obtained by combining ideas from previous results in the bandits literature and the recent work in replicable learning of (ILPS22).

6.1.2 Warm-up II: Bandit Algorithms and Replicable Mean Estimation

First, we remark that we work in the stochastic setting and the distributions of the rewards of the two arms are subgaussian. Thus, the problem of estimating their mean is an instance of a statistical query for which we can use the algorithm of (ILPS22) to get a replicable mean estimator for the distributions of the rewards of the arms.

Proposition 6.1.1 (Replicable Mean Estimation (ILPS22))**.** *Let $\tau, \delta, \rho \in [0,1]$. There exists a ρ-replicable algorithm* `ReprMeanEstimation` *that draws $\Omega\left(\frac{\log(1/\delta)}{\tau^2(\rho-\delta)^2}\right)$ samples from a distribution with mean μ and computes an estimate $\widehat{\mu}$ that satisfies $|\widehat{\mu} - \mu| \le \tau$ with probability at least $1 - \delta$.*

Notice that we are working in the regime where $\delta \ll \rho$, so the sample complexity is $\Omega\left(\frac{\log(1/\delta)}{\tau^2\rho^2}\right)$. The straightforward approach is to try to use an optimal multi-armed algorithm for the stochastic setting, such as UCB or arm-elimination (EDMMM06), combined with the replicable mean estimator. However, it is not hard to see that this approach does not give meaningful results: if we want to achieve replicability ρ we need to call the replicable mean estimator routine with parameter $\rho/(KT)$, due to the union bound that we need to take. This means that we need to pull every arm at least K^2T^2 times, so the regret guarantee becomes vacuous. This gives us the first key insight to tackle the problem: we need to reduce the number of calls to the mean estimator. Hence, we will draw inspiration from the line of work in stochastic batched[1] bandits (GHRZ19; EKMM21) to derive *replicable* bandit algorithms.

[1] While sequential bandit problems have been studied for almost a century, there is

6.2 Replicable Mean Estimation for Batched Bandits

As a first step, we would like to show how one could combine the existing replicable algorithms of (ILPS22) with the batched bandits approach of (EKMM21) to get some preliminary non-trivial results. We build an algorithm for the K-arm setting, where the gaps Δ_j are unknown to the learner. Let δ be the confidence parameter of the arm elimination algorithm and ρ be the replicability guarantee we want to achieve. Our approach is the following: let us, deterministically, split the time interval into sub-intervals of increasing length. We treat each sub-interval as a batch of samples where we pull each active arm the same number of times and use the replicable mean estimation algorithm to, empirically, compute the true mean. At the end of each batch, we decide to eliminate some arm j using the standard UCB estimate. Crucially, if we condition on the event that all the calls to the replicable mean estimator return the same number, then the algorithm we propose is replicable.

Algorithm 8 Mean-Estimation Based Replicable Algorithm for Stochastic MAB (Theorem 6.2.1)

```
Input:  time horizon $T$, number of arms $K$, replicability $\rho$
Initialization: $B \leftarrow \log(T)$, $q \leftarrow T^{1/B}$, $c_0 \leftarrow 0$, $\mathcal{A} \leftarrow [K]$, $r \leftarrow T$, $\widehat{\mu}_a \leftarrow 0, \forall a \in \mathcal{A}$
for $i = 1$ to $B - 1$ do
    if $\lfloor q^i \rfloor \cdot |\mathcal{A}| > r$ then
        break
    $c_i = c_{i-1} + \lfloor q^i \rfloor$
    Pull every arm $a \in \mathcal{A}$ for $\lfloor q^i \rfloor$ times
    for $a \in \mathcal{A}$ do
        $\widehat{\mu}_a \leftarrow$ ReprMeanEst($\delta = \frac{1}{2KTB}, \tau = \min\{1, \sqrt{\log(2KTB)/c_i}\}, \rho' = \frac{\rho}{KB}$)    ▷ Proposition 6.1.1
    $r \leftarrow r - |\mathcal{A}| \cdot \lfloor q^i \rfloor$
    for $a \in \mathcal{A}$ do
        if $\widehat{\mu}_a < \max_{a \in \mathcal{A}} \widehat{\mu}_a - 2\tau$ then
            Remove $a$ from $\mathcal{A}$
In the last batch play the arm from $\mathcal{A}$ with the smallest index
```

much interest in the batched setting too. In many settings, like medical trials, one has to take a lot of actions in parallel and observe their rewards later. The works of (AO10) and (CBDS13) provided sequential bandit algorithms which can easily work in the batched setting. The works of (GHRZ19) and (EKMM21) are focusing exclusively on the batched setting.

Theorem 6.2.1. *Let $T \in \mathbb{N}, \rho \in (0,1]$. There exists a ρ-replicable algorithm (presented in Algorithm 8) for the stochastic bandit problem with K arms and gaps $(\Delta_j)_{j\in[K]}$ whose expected regret is*

$$\mathbb{E}[R_T] \leq C \cdot \frac{K^2 \log^2(T)}{\rho^2} \sum_{j:\Delta_j>0} \left(\Delta_j + \frac{\log(KT\log(T))}{\Delta_j} \right) ,$$

where $C > 0$ is an absolute numerical constant, and its running time is polynomial in K, T and $1/\rho$.

The above result, whose proof can be found in Section 6.5, states that, by combining the tools from (ILPS22) and (EKMM21), we can design a replicable bandit algorithm with (instance-dependent) expected regret $O(K^2 \log^3(T)/\rho^2)$. Notice that the regret guarantee has an extra $K^2 \log^2(T)/\rho^2$ factor compared to its non-replicable counterpart in (EKMM21) (Theorem 5.1). This is because, due to a union bound over the rounds and the arms, we need to call the replicable mean estimator with parameter $\rho/(K\log(T))$. In the next section, we show how to get rid of the $\log^2(T)$ by designing a new algorithm.

6.3 Improved Algorithms for Replicable Stochastic Bandits

While the previous result provides a non-trivial regret bound, it is not optimal with respect to the time horizon T. In this section, we show to improve it by designing a new algorithm, presented in Algorithm 9, which satisfies the guarantees of Theorem 6.3.1 and, essentially, decreases the dependence on the time horizon T from $\log^3(T)$ to $\log(T)$. Our main result for replicable stochastic multi-armed bandits with K arms follows.

Theorem 6.3.1. *Let $T \in \mathbb{N}, \rho \in (0,1]$. There exists a ρ-replicable algorithm (presented in Algorithm 9) for the stochastic bandit problem with K arms and gaps $(\Delta_j)_{j\in[K]}$ whose expected regret is*

$$\mathbb{E}[R_T] \leq C \cdot \frac{K^2}{\rho^2} \sum_{j:\Delta_j>0} \left(\Delta_j + \frac{\log(KT\log(T))}{\Delta_j} \right) ,$$

where $C > 0$ is an absolute numerical constant, and its running time is polynomial in K, T and $1/\rho$.

Note that, compared to the non-replicable setting, we incur an extra factor of K^2/ρ^2 in the regret. The proof can be found in Section 6.6. Let us now describe how Algorithm 9 works. We decompose the time horizon into $B = \log(T)$ batches. Without the replicability constraint, one could draw q^i samples in batch i from

Algorithm 9 Replicable Algorithm for Stochastic Multi-Armed Bandits (Theorem 6.3.1)

```
Input: time horizon $T$, number of arms $K$, replicability $\rho$
Initialization: $B \leftarrow \log(T)$, $q \leftarrow T^{1/B}$, $c_0 \leftarrow 0$, $\mathcal{A}_0 \leftarrow [K]$, $r \leftarrow T$, $\widehat{\mu}_a \leftarrow 0, \forall a \in \mathcal{A}_0$
$\beta \leftarrow \lfloor \max\{K^2/\rho^2, 2304\} \rfloor$
for $i = 1$ to $B - 1$ do
    if $\beta \lfloor q^i \rfloor \cdot |\mathcal{A}_i| > r$ then
        break
    $\mathcal{A}_i \leftarrow \mathcal{A}_{i-1}$
    for $a \in \mathcal{A}_i$ do
        Pull arm $a$ for $\beta \lfloor q^i \rfloor$ times
        Compute the empirical mean $\widehat{\mu}_\alpha^{(i)}$
    $c_i \leftarrow c_{i-1} + \lfloor q^i \rfloor$
    $\widetilde{c_i} \leftarrow \beta c_i$
    $\widetilde{U}_i \leftarrow \sqrt{2\log(2KTB)/\widetilde{c}_i}$
    $U_i \leftarrow \sqrt{2\log(2KTB)/c_i}$
    $\overline{U}_i \leftarrow \mathrm{Uni}[U_i/2, U_i]$
    $r \leftarrow r - \beta \cdot |\mathcal{A}_i| \cdot \lfloor q^i \rfloor$
    for $a \in \mathcal{A}_i$ do
        if $\widehat{\mu}_a^{(i)} + \widetilde{U}_i < \max_{a \in \mathcal{A}_i} \widehat{\mu}_a^{(i)} - \overline{U}_i$ then
            Remove $a$ from $\mathcal{A}_i$
In the last batch play the arm from $\mathcal{A}_{B-1}$ with the smallest index
```

each arm and estimate the mean reward. With the replicability constraint, we have to boost this: in each batch i, we pull each active arm $O(\beta q^i)$ times, for some q to be determined, where $\beta = O(K^2/\rho^2)$ is the replicability blow-up. Using these samples, we compute the empirical mean $\widehat{\mu}_\alpha^{(i)}$ for any active arm α. Note that $\widetilde{U}_i$ in Algorithm 9 corresponds to the size of the actual confidence interval of the estimation and U_i corresponds to the confidence interval of an algorithm that does not use the β-blow-up in the number of samples. The novelty of our approach comes from the choice of the interval around the mean of the maximum arm: we pick a threshold uniformly at random from an interval of size $U_i/2$ around the maximum mean. Then, the algorithm checks whether $\widehat{\mu}_a^{(i)} + \widetilde{U}_i < \max \widehat{\mu}_{a'}^{(i)} - \overline{U}_i$, where max runs over the active arms a' in batch i, and eliminates arms accordingly. To prove the result we show that there are three regions that some arm j can be in relative to the confidence interval of the best arm in batch i (cf. Section 6.6). If it lies in two of these regions, then the decision of whether to keep it or discard it is the same in both executions of the algorithm. However, if it is in the third region, the decision could be different between parallel executions, and since it relies on some external and unknown randomness, it is not clear how to reason about it. To overcome this issue, we use the random threshold to argue about the probability that the decision between two executions differs. The crucial observation that allows us to get rid of the extra $\log^2(T)$ factor is that there are correlations between consecutive batches: we prove that if some arm j lies in this "bad" region in some batch i, then it will be outside this region after a constant number of batches.

6.4 Replicable Stochastic Linear Bandits

We now investigate replicability in the more general setting of stochastic linear bandits. In this setting, each arm is a vector $a \in \mathbb{R}^d$ belonging to some action set $\mathcal{A} \subseteq \mathbb{R}^d$, and there is a parameter $\theta^\star \in \mathbb{R}^d$ unknown to the player. In round t, the player chooses some action $a_t \in \mathcal{A}$ and receives a reward $r_t = \langle \theta^\star, a_t \rangle + \eta_t$, where η_t is a zero-mean 1-subgaussian random variable independent of any other source of randomness. This means that $\mathbb{E}[\eta_t] = 0$ and satisfies $\mathbb{E}[\exp(\lambda \eta_t)] \le \exp(\lambda^2/2)$ for any $\lambda \in \mathbb{R}$. For normalization purposes, it is standard to assume that $\|\theta^\star\|_2 \le 1$ and $\sup_{a \in \mathcal{A}} \|a\|_2 \le 1$. In the linear setting, the expected regret after T pulls $a_1, \ldots, a_T$ can be written as

$$\mathbb{E}[R_T] = T \sup_{a \in \mathcal{A}} \langle \theta^\star, a \rangle - \mathbb{E}\left[\sum_{t=1}^{T} \langle \theta^\star, a_t \rangle\right] .$$

In Section 6.4.1 we provide results for the finite action space case, i.e., when $|\mathcal{A}| = K$. Next, in Section 6.4.2, we study replicable linear bandit algorithms when dealing with infinite action spaces. In the following, we work in the regime where $T \gg d$. We underline that our approach leverages connections of stochastic linear bandits with G-optimal experiment design, core sets constructions, and least-squares estimators. Roughly speaking, the goal of G-optimal design is to find a (small) subset of arms $\mathcal{A}'$, which is called the core set, and define a distribution π over them with the following property: for any $\varepsilon > 0, \delta > 0$ pulling only these arms for an appropriate number of times and computing the least-squares estimate $\widehat{\theta}$ guarantees that $\sup_{a \in \mathcal{A}} \langle a, \theta^* - \widehat{\theta} \rangle \le \varepsilon$, with probability $1 - \delta$. For an extensive discussion, we refer to Chapters 21 and 22 of (LS20).

6.4.1 Finite Action Set

We first introduce a lemma that allows us to reduce the size of the action set that our algorithm has to search over.

Lemma 6.4.1 (See Chapters 21 and 22 in (LS20))**.** *For any finite action set $\mathcal{A}$ that spans $\mathbb{R}^d$ and any $\delta, \varepsilon > 0$, there exists an algorithm that, in time polynomial in d, computes a multi-set of $\Theta(d \log(1/\delta)/\varepsilon^2 + d \log \log d)$ actions (possibly with repetitions) such that (i) they span $\mathbb{R}^d$ and (ii) if we perform these actions in a* *batched stochastic d-dimensional linear bandits setting with true parameter $\theta^\star \in \mathbb{R}^d$ and let $\widehat{\theta}$ be the least-squares estimate for $\theta^\star$, then, for any $a \in \mathcal{A}$, with probability at least $1-\delta$, we have $\left|\left\langle a, \theta^\star - \widehat{\theta}\right\rangle\right| \leq \varepsilon$.*

Essentially, the multi-set in Lemma 6.4.1 is obtained using an approximate *G-optimal design* algorithm. Thus, it is crucial to check whether this can be done in a replicable manner. Recall that the above set of distinct actions is called the core set and is the solution of an (approximate) G-optimal design problem. To be more specific, consider a distribution $\pi : \mathcal{A} \to [0,1]$ and define $V(\pi) = \sum_{a\in\mathcal{A}} \pi(a)aa^\top \in \mathbb{R}^{d\times d}$ and $g(\pi) = \sup_{a\in\mathcal{A}} \|a\|^2_{V(\pi)^{-1}}$. The distribution π is called a design and the goal of G-optimal design is to find a design that minimizes g. Since the number of actions is finite, this problem reduces to an optimization problem which can be solved efficiently using standard optimization methods (e.g., the Frank-Wolfe method). Since the initialization is the same, the algorithm that finds the optimal (or an approximately optimal) design is replicable under the assumption that the gradients and the projections do not have numerical errors. This perspective is orthogonal to the work of (AJJ+22), that defines replicability from a different viewpoint.

Algorithm 10 Replicable Algorithm for Stochastic Linear Bandits (Theorem 6.4.2)

```
Input: number of arms $K$, time horizon $T$, replicability $\rho$
Initialization: $B \leftarrow \log(T)$, $q \leftarrow (T/c)^{1/B}$, $\mathcal{A} \leftarrow [K]$, $r \leftarrow T$
$\beta \leftarrow \lfloor \max\{K^2/\rho^2, 2304\} \rfloor$
for $i = 1$ to $B-1$ do
    $\widetilde{\varepsilon}_i = \sqrt{d\log(KT^2)/(\beta q^i)}$
    $\varepsilon_i = \sqrt{d\log(KT^2)/q^i}$
    $n_i = 10d\log(KT^2)/\varepsilon_i^2$
    $a_1, \ldots, a_{n_i} \leftarrow$ multi-set given by Lemma 6.4.1 with parameters $\delta = 1/(KT^2)$ and $\varepsilon = \widetilde{\varepsilon}_i$
    if $n_i > r$ then
        break
    Pull every arm $a_1, \ldots, a_{n_i}$ and receive rewards $r_1, \ldots, r_{n_i}$
    Compute the LSE $\widehat{\theta}_i \leftarrow \left(\sum_{j=1}^{n_i} a_j a_j^T\right)^{-1} \left(\sum_{j=1}^{n_i} a_j r_j\right)$
    $\bar{\varepsilon}_i \leftarrow \mathrm{Uni}[\varepsilon_i/2, \varepsilon_i]$
    $r \leftarrow r - n_i$
    for $a \in \mathcal{A}$ do
        if $\langle a, \widehat{\theta}_i\rangle + \widetilde{\varepsilon}_i < \max_{a\in\mathcal{A}}\langle a, \widehat{\theta}_i\rangle - \bar{\varepsilon}_i$ then
            Remove $a$ from $\mathcal{A}$
In the last batch play $\arg\max_{a\in\mathcal{A}}\langle a, \widehat{\theta}_{B-1}\rangle$
```

In our batched bandit algorithm (Algorithm 10), the multi-set of arms $a_1, \ldots, a_{n_i}$

computed in each batch is obtained via a deterministic algorithm with runtime poly(K, d), where $|\mathcal{A}| = K$. Hence, the multi-set will be the same in two different executions of the algorithm. On the other hand, the LSE will not be since it depends on the stochastic rewards. We apply the techniques that we developed in the replicable stochastic MAB setting in order to design our algorithm. Our main result for replicable d-dimensional stochastic linear bandits with K arms follows. For the proof, we refer to Section 6.7.

Theorem 6.4.2. *Let $T \in \mathbb{N}, \rho \in (0, 1]$. There exists a ρ-replicable algorithm (presented in Algorithm 10) for the stochastic d-dimensional linear bandit problem with K arms whose expected regret is*

$$\mathbb{E}[R_T] \leq C \cdot \frac{K^2}{\rho^2} \sqrt{dT \log(KT)}\,,$$

where $C > 0$ is an absolute numerical constant, and its running time is polynomial in d, K, T and $1/\rho$.

Note that the best known non-replicable algorithm achieves an upper bound of $\widetilde{O}(\sqrt{dT \log(K)})$ and, hence, our algorithm incurs a replicability overhead of order K^2/ρ^2. The intuition behind the proof is similar to the multi-armed bandit setting in Section 6.3.

6.4.2 Infinite Action Set

Let us proceed to the setting where the action set $\mathcal{A}$ is unbounded. Unfortunately, even when $d = 1$, we cannot directly get an algorithm that has satisfactory regret guarantees by discretizing the space and using Algorithm 10. The approach of (EKMM21) is to discretize the action space and use an $1/T$-net to cover it, i.e. a set $\mathcal{A}' \subseteq A$ such that for all $a \in \mathcal{A}$ there exists some $a' \in \mathcal{A}'$ with $||a - a'||_2 \leq 1/T$. It is known that there exists such a net of size at most $(3T)^d$ (Ver18, Corollary 4.2.13). Then, they apply the algorithm for the finite arms setting, increasing their regret guarantee by a factor of $\sqrt{d}$. However, our replicable algorithm for this setting contains an additional factor of K^2 in the regret bound. Thus, even when $d = 1$, our regret guarantee is greater than T, so the bound is vacuous. One way to fix this issue and get a sublinear regret guarantee is to use a smaller net. We use a $1/T^{1/(4d+2)}-$net that has size at most $(3T)^{\frac{d}{4d+2}}$ and this yields an expected regret of order $O(T^{4d+1/(4d+2)}\sqrt{d \log(T)}/\rho^2)$. For further details, we refer to Section 6.8.

Even though the regret guarantee we managed to get using the smaller net of Section 6.8 is sublinear in T, it is not a satisfactory bound. The next step is to provide an algorithm for the infinite action setting using a replicable LSE subroutine combined with the batching approach of (EKMM21). We will make use of the next lemma.

Lemma 6.4.3 (Section 21.2 Note 3 of (LS20))**.** *There exists a deterministic algorithm that, given an action space $\mathcal{A} \subseteq \mathbb{R}^d$, computes a 2-approximate G-optimal design π with a core set of size $O(d \log\log(d))$.*

We additionally prove the next useful lemma, which, essentially, states that we can assume without loss of generality that every arm in the support of π has mass at least $\Omega(1/(d\log(d)))$. We refer to Section 6.10.1 for the proof.

Lemma 6.4.4 (Effective Support)**.** *Let π be the distribution that corresponds to the 2-approximate optimal G-design of Lemma 6.4.3 with input $\mathcal{A}$. Assume that $\pi(a) \leq c/(d\log(d))$, where $c > 0$ is some absolute numerical constant, for some arm a in the core set. Then, we can construct a distribution $\widehat{\pi}$ such that, for any arm a in the core set, $\widehat{\pi}(a) \geq C/(d\log(d))$, where $C > 0$ is an absolute constant, so that it holds*

$$\sup_{a' \in \mathcal{A}} \|a'\|^2_{V(\widehat{\pi})^{-1}} \leq 4d\,.$$

The upcoming lemma is a replicable algorithm for the least-squares estimator and, essentially, builds upon Lemma 6.4.3 and Lemma 6.4.4. Its proof can be found at Section 6.10.2. We believe that this technical result could be interesting on its own since it can be applied to other problems as well.

Lemma 6.4.5 (Replicable LSE)**.** *Let $\rho, \varepsilon \in (0, 1]$ and $0 < \delta \leq \min\{\rho, 1/d\}$*[2]*. Consider an environment of d-dimensional stochastic linear bandits with infinite action space $\mathcal{A}$. Assume that π is a 4-approximate optimal design with associated core set $\mathcal{C}$ as computed by Lemma 6.4.3 with input $\mathcal{A}$. There exists a ρ-replicable algorithm that pulls each arm $a \in \mathcal{C}$ a total of*

$$\Omega\left(\frac{d^4 \log(d/\delta) \log^2 \log(d) \log\log\log(d)}{\varepsilon^2 \rho^2}\right)$$

times and outputs an estimate θ_{SQ} that satisfies $\sup_{a \in \mathcal{A}} |\langle a, \theta_{\mathrm{SQ}} - \theta^\star \rangle| \leq \varepsilon$, with probability at least $1 - \delta$.

The main result for the infinite actions' case, obtained by Algorithm 11, follows. Its proof can be found at Section 6.9.

Theorem 6.4.6. *Let $T \in \mathbb{N}, \rho \in (0, 1]$. There exists a ρ-replicable algorithm (presented in Algorithm 11) for the stochastic d-dimensional linear bandit problem with infinite action set whose expected regret is*

$$\mathbb{E}[R_T] \leq C \cdot \frac{d^4 \log(d) \log^2 \log(d) \log\log\log(d)}{\rho^2} \sqrt{T} \log^{3/2}(T)\,,$$

where $C > 0$ is an absolute numerical constant, and its running time is polynomial in T^d and $1/\rho$.

[2] We can handle the case of $0 < \delta \leq d$ by paying an extra $\log d$ factor in the sample complexity.

Our algorithm for the infinite arm linear bandit case enjoys an expected regret of order $\widetilde{O}(\text{poly}(d)\sqrt{T})$. We underline that the dependence of the regret on the time horizon is (almost) optimal, and we incur an extra d^3 factor in the regret guarantee compared to the non-replicable algorithm of (EKMM21). We now comment on the time complexity of our algorithm.

Remark 6. *The current implementation of our algorithm requires time exponential in d. However, for a general convex set $\mathcal{A}$, given access to a separation oracle for it and an oracle that computes an (approximate) G-optimal design, we can execute it in polynomial time and with polynomially many calls to the oracle. Notably, when $\mathcal{A}$ is a polytope such oracles exist. We underline that computational complexity issues also arise in the traditional setting of linear bandits with an infinite number of arms and the computational overhead that the replicability requirement adds is minimal. For further details, we refer to Section 6.11.*

6.5 The Proof of Theorem 6.2.1

Theorem. *Let $T \in \mathbb{N}, \rho \in (0,1]$. There exists a ρ-replicable algorithm (presented in Algorithm 8) for the stochastic bandit problem with K arms and gaps $(\Delta_j)_{j\in[K]}$ whose expected regret is*

$$\mathbb{E}[R_T] \le C \cdot \frac{K^2 \log^2(T)}{\rho^2} \sum_{j:\Delta_j>0} \left(\Delta_j + \frac{\log(2KT\log(T))}{\Delta_j}\right),$$

where $C > 0$ is an absolute numerical constant, and its running time is polynomial in K, T and $1/\rho$.

Proof. First, we claim that the algorithm is ρ-replicable: since the elimination decisions are taken in the same iterates and are based solely on the mean estimations, the replicability of the algorithm of Proposition 6.1.1 implies the replicability of the whole algorithm. In particular,

$$\mathbf{Pr}[(a_1, ..., a_T) \neq (a'_1, ..., a'_T)] = \mathbf{Pr}[\exists i \in [B], \exists j \in [K] : \widehat{\mu}_j^{(i)} \text{ was not replicable}] \le \rho\,.$$

During each batch i, we draw for any active arm $\lfloor q^i \rfloor$ fresh samples for a total of c_i samples and use the replicable mean estimation algorithm to estimate its mean. For an active arm, at the end of some batch $i \in [B]$, we say that its estimation is "correct" if the estimation of its mean is within $\sqrt{\log(2KTB)/c_i}$ from the true mean. Using Proposition 6.1.1, the estimation of any active arm at the end of any batch (except possibly the last batch) is correct with probability at least $1 - 1/(2KTB)$ and so, by the union bound, the probability that the estimation is incorrect for some arm at the end of some batch is bounded by $1/T$. We remark that when $\delta < \rho$, the sample complexity of Proposition 6.1.1 reduces to

$O(\log(1/\delta)/(\tau^2\rho^2))$. Let $\mathcal{E}$ denote the event that our estimates are correct. The total expected regret can be bounded as

$$\mathbb{E}[R_T] \le T \cdot 1/T + \mathbb{E}[R_T|\mathcal{E}]\,.$$

It suffices to bound the second term of the RHS and hence we can assume that each gap is correctly estimated within an additive factor of $\sqrt{\log(2KTB)/c_i}$ after batch i. First, due to the elimination condition, we get that the best arm is never eliminated. Next, we have that

$$\mathbb{E}[R_T|\mathcal{E}] = \sum_{j:\Delta_j>0} \Delta_j \,\mathbb{E}[T_j|\mathcal{E}]\,,$$

where T_j is the total number of pulls of arm j. Fix a sub-optimal arm j and assume that $i+1$ was the last batch it was active. Since this arm is not eliminated at the end of batch i, and the estimations are correct, we have that

$$\Delta_j \le \sqrt{\log(2KTB)/c_i}\,,$$

and so $c_i \le \log(2KTB)/\Delta_j^2$. Hence, the number of pulls to get the desired bound due to Proposition 6.1.1 is (since we need to pull an arm c_i/ρ_1^2 times in order to get an estimate at distance $\sqrt{\log(1/\delta)/c_i^2}$ with probability $1-\delta$ in a ρ_1-replicable manner when $\delta < \rho_1$)

$$T_j \le c_{i+1}/\rho_1^2 = q/\rho_1^2(1+c_i) \le q/\rho_1^2 \cdot (1 + \log(2KTB)/\Delta_j^2)\,.$$

This implies that the total regret is bounded by

$$\mathbb{E}[R_T] \le 1 + q/\rho_1^2 \cdot \sum_{j:\Delta_j>0} \left(\Delta_j + \frac{\log(2KTB)}{\Delta_j}\right)\,.$$

We finally set $q = T^{1/B}$ and $B = \log(T)$. Moreover, we have that $\rho_1 = \rho/(KB)$. These yield

$$\mathbb{E}[R_T] \le \frac{K^2\log^2(T)}{\rho^2} \sum_{j:\Delta_j>0} \left(\Delta_j + \frac{\log(2KT\log(T))}{\Delta_j}\right)\,.$$

This completes the proof. □

6.6 The Proof of Theorem 6.3.1

Theorem. *Let $T \in \mathbb{N}, \rho \in (0,1]$. There exists a ρ-replicable algorithm (presented in Algorithm 9) for the stochastic bandit problem with K arms and gaps $(\Delta_j)_{j\in[K]}$ whose expected regret is*

$$\mathbb{E}[R_T] \le C \cdot \frac{K^2}{\rho^2} \sum_{j:\Delta_j>0} (\Delta_j + \log(KT\log(T))/\Delta_j)\ ,$$

for some absolute numerical constant $C > 0$, *and its running time is polynomial in* K, T *and* $1/\rho$.

To give some intuition, we begin with a non tight analysis which, however, provides the main ideas behind the actual proof.

Non Tight Analysis Assume that the environment has K arms with unknown means μ_i and let T be the number of rounds. Consider B to the total number of batches and $\beta > 1$. We set $q = T^{1/B}$. In each batch $i \in [B]$, we pull each arm $\beta \lfloor\ q^i \rfloor$ times. Hence, after the i-th batch, we will have drawn $\widetilde{c}_i = \sum_{1 \le j \le i} \beta \lfloor q^j \rfloor$ independent and identically distributed samples from each arm. Let us also set $c_i = \sum_{1 \le j \le i} \lfloor q^j \rfloor$.

Let us fix $i \in [B]$. Using Hoeffding's bound for subgaussian concentration, the length of the confidence bound for arm $j \in [K]$ that guarantees $1 - \delta$ probability of success (in the sense that the empirical estimate $\widehat{\mu}_j$ will be close to the true μ_j) is equal to

$$\widetilde{U}_i = \sqrt{2 \log(1/\delta)/\widetilde{c}_i}\,,$$

when the estimator uses $\widetilde{c}_i$ samples. Also, let

$$U_i = \sqrt{2 \log(1/\delta)/c_i}\,.$$

Assume that the active arms at the batch iteration i lie in the set $\mathcal{A}_i$. Consider the estimates $\{\widehat{\mu}_j^{(i)}\}_{i \in [B], j \in \mathcal{A}_i}$, where $\widehat{\mu}_j^{(i)}$ is the empirical mean of arm j using $\widetilde{c}_i$ samples. We will eliminate an arm j at the end of the batch iteration i if

$$\widehat{\mu}_j^{(i)} + \widetilde{U}_i \le \max_{t \in \mathcal{A}_i} \widehat{\mu}_t^{(i)} - \overline{U}_i\,,$$

where $\overline{U}_i \sim \mathrm{Uni}[U_i/2, U_i]$. For the remaining of the proof, we condition on the event $\mathcal{E}$ that for every arm $j \in [K]$ and every batch $i \in [B]$ the true mean is within $\widetilde{U}_i$ from the empirical one.

We first argue about the replicability of our algorithm. Consider a fixed round i (end of i-th batch) and a fixed arm j. Let $i^\star$ be the optimal empirical arm after the i-th batch.

Let $\widehat{\mu}_j^{(i)'}, \widehat{\mu}_{i^\star}^{(i)'}$ the empirical estimates of arms $j, i^\star$ after the i-th batch, under some other execution of the algorithm. We condition on the event $\mathcal{E}'$ for the other execution as well. Notice that $|\widehat{\mu}_j^{(i)'} - \widehat{\mu}_j^{(i)}| \le 2\widetilde{U}_i, |\widehat{\mu}_{i^\star}^{(i)'} - \widehat{\mu}_{i^\star}^{(i)}| \le 2\widetilde{U}_i$. Notice that, since the randomness of $\overline{U}_i$ is shared, if $\widehat{\mu}_j^{(i)} + \widetilde{U}_i \ge \widehat{\mu}_{i^\star}^{(i)} - \overline{U}_i + 4\widetilde{U}_i$, then the arm j will not be eliminated after the i-th batch in some other execution of the algorithm as well. Similarly, if $\widehat{\mu}_j^{(i)} + \widetilde{U}_i < \widehat{\mu}_{i^\star}^{(i)} - \overline{U}_i - 4\widetilde{U}_i$ the the arm j will get eliminated after the i-th batch in some other execution of the algorithm as well. In particular, this means that if $\widehat{\mu}_j^{(i)} - 2\widetilde{U}_i > \widehat{\mu}_{i^\star}^{(i)} + \widetilde{U}_i - U_i/2$ then the arm j will not get eliminated in some other execution of the algorithm and if $\widehat{\mu}_j^{(i)} + 5\widetilde{U}_i < \widehat{\mu}_{i^\star}^{(i)} - U_i$ then the arm j will also get eliminated in some other execution of the algorithm with probability

1 under the event $\mathcal{E} \cap \mathcal{E}'$. We call the above two cases good since they preserve replicability. Thus, it suffices to bound the probability that the decision about arm j will be different between the two executions when we are in neither of these cases. Then, the worst case bound due to the mass of the uniform probability measure is

$$\frac{16\sqrt{2\log(1/\delta)/\widetilde{c}_i}}{\sqrt{2\log(1/\delta)/c_i}}\,.$$

This implies that the probability mass of the bad event is at most $16\sqrt{c_i/\widetilde{c}_i} = 16\sqrt{1/\beta}$. A union bound over all arms and batches yields that the probability that two distinct executions differ in at least one pull is

$$\mathbf{Pr}[(a_1,\ldots,a_T) \neq (a'_1,\ldots,a'_T)] \leq 16KB\sqrt{1/\beta} + 2\delta\,,$$

and since $\delta \leq \rho$ it suffices to pick $\beta = 768K^2B^2/\rho^2$.

We now focus on the regret of our algorithm. Let us set $\delta = 1/(KTB)$. Fix a sub-optimal arm j and assume that batch $i+1$ was the last batch that is was active. We obtain that the total number of pulls of this arm is

$$T_j \leq \widetilde{c}_{i+1} \leq \beta q(1+c_i) \leq \beta q(1 + 8\log(1/\delta)/\Delta_j^2)$$

From the replicability analysis, it suffices to take β of order $K^2\log^2(T)/\rho^2$ and so

$$\mathbb{E}[R_T] \leq T \cdot 1/T + \mathbb{E}[R_T|\mathcal{E}] = 1 + \sum_{j:\Delta_j>0} \Delta_j\,\mathbb{E}[T_j|\mathcal{E}] \leq \frac{C\cdot K^2\log^2(T)}{\rho^2}\sum_{j:\Delta_j>0}\left(\Delta_j + \frac{\log(KT\log(T))}{\Delta_j}\right),$$

for some absolute constant $C > 0$.

Notice that the above analysis, which uses a naive union bound, does not yield the desired regret bound. We next provide a more tight analysis of the same algorithm that achieves the regret bound of Theorem 6.3.1.

Improved Analysis (***The Proof of Theorem 6.3.1***) In the previous analysis, we used a union bound over all arms and all batches in order to control the probability of the bad event. However, we can obtain an improved regret bound as follows. Fix a sub-optimal arm $i \in [K]$ and let t be the first round that it appears in the bad event. We claim that after a constant number of rounds, this arm will be eliminated. This will shave the $O(\log^2(T))$ factor from the regret bound. Essentially, as indicated in the previous proof, the bad event corresponds to the case where the randomness of the cut-off threshold $\overline{U}$ can influence the decision of whether the algorithm eliminates an arm or not. The intuition is that during the rounds t and $t+1$, given that the two intervals intersected at round t, we know that the probability that they intersect again is quite small since the interval of the optimal mean is moving upwards, the interval of the sub-optimal mean is concentrating around the guess and the two estimations have been moved by at most a constant times the interval's length.

Since the bad event occurs at round t, we know that

$$\widehat{\mu}_j^{(t)} \in \left[\widehat{\mu}_{t^\star}^{(t)} - U_t - 5\widetilde{U}_t, \widehat{\mu}_{t^\star}^{(t)} - U_t/2 + 3\widetilde{U}_t\right].$$

In the above $\widehat{\mu}_{t^\star}^t$ is the estimate of the optimal mean at round t whose index is denoted by $t^\star$. Now assume that the bad event for arm j also occurs at round $t+k$. Then, we have that

$$\widehat{\mu}_j^{(t+k)} \in \left[\widehat{\mu}_{(t+k)^\star}^{(t+k)} - U_{t+k} - 5\widetilde{U}_{t+k}, \widehat{\mu}_{(t+k)^\star}^{(t+k)} - U_{t+k}/2 + 3\widetilde{U}_{t+k}\right].$$

First, notice that since the concentration inequality under event $\mathcal{E}$ holds for rounds $t, t+k$ we have that $\widehat{\mu}_j^{(t+k)} \leq \widehat{\mu}_j^{(t)} + \widetilde{U}_t + \widetilde{U}_{t+k}$. Thus, combining it with the above inequalities gives us

$$\widehat{\mu}_{(t+k)^\star}^{(t+k)} - U_{t+k} - 5\widetilde{U}_{t+k} \leq \widehat{\mu}_j^{(t+k)} \leq \widehat{\mu}_j^{(t)} + \widetilde{U}_t + \widetilde{U}_{t+k} \leq \widehat{\mu}_{t^\star}^{(t)} - U_t/2 + 4\widetilde{U}_t + \widetilde{U}_{t+k}.$$

We now compare $\widehat{\mu}_{t^\star}^{(t)}, \widehat{\mu}_{(t+k)^\star}^{(t+k)}$. Let o denote the optimal arm. We have that

$$\widehat{\mu}_{(t+k)^\star}^{(t+k)} \geq \widehat{\mu}_o^{(t+k)} \geq \mu_o - \widetilde{U}_{t+k} \geq \mu_{t^\star} - \widetilde{U}_{t+k} \geq \widehat{\mu}_{t^\star}^{(t)} - \widetilde{U}_t - \widetilde{U}_{t+k}.$$

This gives us that

$$\widehat{\mu}_{t^\star}^{(t)} - U_{t+k} - 6\widetilde{U}_{t+k} - \widetilde{U}_t \leq \widehat{\mu}_{(t+k)^\star}^{(t+k)} - U_{t+k} - 5\widetilde{U}_{t+k}.$$

Thus, we have established that

$$\begin{aligned}\widehat{\mu}_{t^\star}^{(t)} - U_{t+k} - 6\widetilde{U}_{t+k} - \widetilde{U}_t \leq \widehat{\mu}_{t^\star}^{(t)} - U_t/2 + 4\widetilde{U}_t + \widetilde{U}_{t+k} &\implies \\ U_{t+k} \geq U_t/2 - 7\widetilde{U}_{t+k} - 5\widetilde{U}_t \geq U_t/2 - 12\widetilde{U}_t.\end{aligned}$$

Since $\beta \geq 2304$, we get that $12\widetilde{U}_t \leq U_t/4$. Thus, we get that

$$U_{t+k} \geq U_t/4.$$

Notice that

$$\frac{U_{t+k}}{U_t} = \sqrt{\frac{c_t}{c_{t+k}}},$$

thus it immediately follows that

$$\begin{aligned}\frac{c_t}{c_{t+k}} \geq \frac{1}{16} \implies \frac{q^{t+1}-1}{q^{t+k+1}-1} \geq \frac{1}{16} \implies 16\left(1 - \frac{1}{q^{t+1}}\right) \geq q^k - \frac{1}{q^{t+1}} \implies \\ q^k \leq 16 + \frac{1}{q^{t+1}} \leq 17 \implies k \log q \leq \log 17 \implies k \leq 5,\end{aligned}$$

when we pick $B = \log(T)$ batches. Thus, for every arm the bad event can happen at most 6 times, by taking a union bound over the K arms we see that the probability that our algorithm is not replicable is at most $O(K\sqrt{1/\beta})$, so picking $\beta = \Theta(K^2/\rho^2)$ suffices to get the result.

6.7 The Proof of Theorem 6.4.2

Theorem. *Let $T \in \mathbb{N}, \rho \in (0,1]$. There exists a ρ-replicable algorithm (presented in Algorithm 10) for the stochastic d-dimensional linear bandit problem with K arms whose expected regret is*

$$\mathbb{E}[R_T] \leq C \cdot \frac{K^2}{\rho^2}\sqrt{dT\log(KT)}\,,$$

for some absolute numerical constant $C > 0$, and its running time is polynomial in d, K, T and $1/\rho$.

Proof. Let c, C be the numerical constants hidden in Lemma 6.4.1, i.e., the size of the multi-set is in the interval $[cd\log(1/\delta)/\varepsilon^2, Cd\log(1/\delta)/\varepsilon^2]$. We know that the size of each batch $n_i \in [cq^i, Cq^i]$ (see Lemma 6.4.1), so by the end of the $B-1$ batch we will have less than n_B pulls left. Hence, the number of batches is at most B.

We first define the event $\mathcal{E}$ that the estimates of all arms after the end of each batch are accurate, i.e., for every active arm a at the beginning of the i-th batch, at the end of the batch we have that $\left|\left\langle a, \widehat{\theta}_i - \theta^\star\right\rangle\right| \leq \widetilde{\varepsilon}_i$. Since $\delta = 1/(KT^2)$ and there are at most T batches and K active arms in each batch, a simple union bound shows that $\mathcal{E}$ happens with probability at least $1 - 1/T$. We condition on the event $\mathcal{E}$ throughout the rest of the proof.

We now argue about the regret bound of our algorithm. We first show that any optimal arm a^* will not get eliminated. Indeed, consider any sub-optimal arm $a \in [K]$ and any batch $i \in [B]$. Under the event $\mathcal{E}$ we have that

$$\langle a, \widehat{\theta}_i\rangle - \langle a^*, \widehat{\theta}_i\rangle \leq (\langle a, \theta^*\rangle + \widetilde{\varepsilon}_i) - (\langle a^*, \theta^*\rangle - \widetilde{\varepsilon}_i) < 2\widetilde{\varepsilon}_i < \varepsilon_i + \overline{\varepsilon}_i.$$

Next, we need to bound the number of times we pull some fixed suboptimal arm $a \in [K]$. We let $\Delta = \langle a^* - a, \theta^*\rangle$ denote the gap and we let i be the smallest integer such that $\varepsilon_i < \Delta/4$. We claim that this arm will get eliminated by the end of batch i. Indeed,

$$\langle a^*, \widehat{\theta}_i\rangle - \langle a, \widehat{\theta}_i\rangle \geq (\langle a^*, \widehat{\theta}_i\rangle - \widetilde{\varepsilon}_i) - (\langle a, \widehat{\theta}_i\rangle + \widetilde{\varepsilon}_i) = \Delta - 2\widetilde{\varepsilon}_i > 4\varepsilon_i - 2\widetilde{\varepsilon}_i > \widetilde{\varepsilon}_i + \overline{\varepsilon}_i.$$

This shows that during any batch i, all the active arms have gap at most $4\varepsilon_{i-1}$. Thus, the regret of the algorithm conditioned on the event $\mathcal{E}$ is at most

$$\sum_{i=1}^{B} 4n_i\varepsilon_{i-1} \leq 4\beta C\sum_{i=1}^{B} q^i\sqrt{d\log(KT^2)/q^{i-1}} \leq 6\beta Cq\sqrt{d\log(KT)}\sum_{i=0}^{B-1} q^{i/2} \leq$$

$$O\left(\beta q^{B/2+1}\sqrt{d\log(KT)}\right) = O\left(\frac{K^2}{\rho^2}q^{B/2+1}\sqrt{d\log(KT)}\right) = O\left(\frac{K^2}{\rho^2}q\sqrt{dT\log(KT)}\right).$$

Thus, the overall regret is bounded by $\delta \cdot T + (1-\delta) \cdot O\left(\frac{K^2}{\rho^2}q\sqrt{dT\log(KT)}\right) = O\left(\frac{K^2}{\rho^2}q\sqrt{dT\log(KT)}\right)$.

We now argue about the replicability of our algorithm. The analysis follows in a similar fashion as in Theorem 6.3.1. Let $\widehat{\theta}_i, \widehat{\theta}'_i$ be the LSE after the i-th batch, under two different executions of the algorithm and assume that the set of active arms. We condition on the event $\mathcal{E}'$ for the other execution as well. Assume that the set of active arms is the same under both executions at the beginning of batch i. Notice that since the set that is guaranteed by Lemma 6.4.1 is computed by a deterministic algorithm, both executions will pull the same arms in batch i. Consider a suboptimal arm a and let $a_{i^*} = \arg\max_{a \in \mathcal{A}} \langle \widehat{\theta}_i, a \rangle, a'_{i^*} = \arg\max_{a \in \mathcal{A}} \langle \widehat{\theta}'_i, a \rangle$. Under the event $\mathcal{E} \cap \mathcal{E}'$ we have that $|\langle a, \widehat{\theta}_i - \widehat{\theta}'_i \rangle| \leq 2\widetilde{\varepsilon}_i, |\langle a_{i^*}, \widehat{\theta}_i - \widehat{\theta}'_i \rangle| \leq 2\widetilde{\varepsilon}_i$, and $|\langle a'_{i^*}, \widehat{\theta}'_i \rangle - \langle a_{i^*}, \widehat{\theta}_i \rangle| \leq 2\widetilde{\varepsilon}_i$. Notice that, since the randomness of $\overline{\varepsilon}_i$ is shared, if $\langle a, \widehat{\theta}_i \rangle + \widetilde{\varepsilon}_i \geq \langle a_{i^*}, \widehat{\theta}_i \rangle - \overline{\varepsilon}_i + 4\widetilde{\varepsilon}_i$, then the arm a will not be eliminated after the i-th batch in some other execution of the algorithm as well. Similarly, if $\langle a, \widehat{\theta}_i \rangle + \widetilde{\varepsilon}_i < \langle a_{i^*}, \widehat{\theta}_i \rangle - \overline{\varepsilon}_i - 4\widetilde{\varepsilon}_i$ the the arm a will get eliminated after the i-th batch in some other execution of the algorithm as well. In particular, this means that if $\langle a, \widehat{\theta}_i \rangle - 2\widetilde{\varepsilon}_i > \langle a_{i^*}, \widehat{\theta}_i \rangle + \widetilde{\varepsilon}_i - \varepsilon_i/2$ then the arm a will not get eliminated in some other execution of the algorithm and if $\langle a, \widehat{\theta}_i \rangle + 5\widetilde{\varepsilon}_i < \langle a_{i^*}, \widehat{\theta}_i \rangle - \varepsilon_i$ then the arm j will also get eliminated in some other execution of the algorithm with probability 1 under the event $\mathcal{E} \cap \mathcal{E}'$. Thus, it suffices to bound the probability that the decision about arm j will be different between the two executions when we are in neither of these cases. Then, the worst case bound due to the mass of the uniform probability measure is

$$\frac{16\sqrt{d\log(1/\delta)/\widetilde{c}_i}}{\sqrt{d\log(1/\delta)/c_i}}\,.$$

This implies that the probability mass of the bad event is at most $16\sqrt{c_i/\widetilde{c}_i} = 16\sqrt{1/\beta}$. A naive union bound would require us to pick $\beta = \Theta(K^2 \log^2 T/\rho^2)$. We next show to avoid the $\log^2 T$ factor. Fix a sub-optimal arm $a \in [K]$ and let t be the first round that it appears in the bad event.

Since the bad event occurs at round t, we know that

$$\langle a, \widehat{\theta}_t \rangle \in \left[\langle a_{t^*}, \widehat{\theta}_t \rangle - \varepsilon_t - 5\widetilde{\varepsilon}_t, \langle a_{t^*}, \widehat{\theta}_t \rangle - \varepsilon_t/2 + 3\widetilde{\varepsilon}_t \right].$$

In the above, a_{t^*} is the optimal arm at round t w.r.t. the LSE. Now assume that the bad event for arm a also occurs at round $t+k$. Then, we have that

$$\langle a, \widehat{\theta}_{t+k} \rangle \in \left[\langle a_{(t+k)^*}, \widehat{\theta}_{t+k} \rangle - \varepsilon_{t+k} - 5\widetilde{\varepsilon}_{t+k}, \langle a_{(t+k)^*}, \widehat{\theta}_{t+k} \rangle - \varepsilon_t/2 + 3\widetilde{\varepsilon}_{t+k} \right].$$

First, notice that since the concentration inequality under event $\mathcal{E}$ holds for rounds $t, t+k$ we have that $\langle a, \widehat{\theta}_{t+k} \rangle \leq \langle a, \widehat{\theta}_t \rangle + \widetilde{\varepsilon}_t + \widetilde{\varepsilon}_{t+k}$. Thus, combining it with the above inequalities gives us

$$\langle a_{(t+k)^*}, \widehat{\theta}_{t+k} \rangle - \varepsilon_{t+k} - 5\widetilde{\varepsilon}_{t+k} \leq \langle a, \widehat{\theta}_{t+k} \rangle \leq \langle a, \widehat{\theta}_t \rangle + \widetilde{\varepsilon}_t + \widetilde{\varepsilon}_{t+k} \leq \langle a_{t^*}, \widehat{\theta}_t \rangle - \varepsilon_t/2 + 4\widetilde{\varepsilon}_t + \widetilde{\varepsilon}_{t+k}.$$

We now compare $\langle a_{t^*}, \widehat{\theta}_t \rangle, \langle a_{(t+k)^*}, \widehat{\theta}_{t+k} \rangle$. Let a^* denote the optimal arm. We have that

$$\langle a_{(t+k)^*}, \widehat{\theta}_{t+k} \rangle \geq \langle a^*, \widehat{\theta}_{t+k} \rangle \geq \langle a^*, \theta^* \rangle - \widetilde{\varepsilon}_{t+k} \geq \langle a_{t^*}, \theta^* \rangle - \widetilde{\varepsilon}_{t+k} \geq \langle a_{t^*}, \widehat{\theta}_t \rangle - \widetilde{\varepsilon}_{t+k} - \widetilde{\varepsilon}_t.$$

This gives us that

$$\langle a_{t^*}, \widehat{\theta}_t \rangle - \varepsilon_{t+k} - 6\widetilde{\varepsilon}_{t+k} - \widetilde{\varepsilon}_t \leq \langle a_{(t+k)^*}, \widehat{\theta}_{t+k} \rangle - \varepsilon_{t+k} - 5\widetilde{\varepsilon}_{t+k}.$$

Thus, we have established that

$$\begin{aligned} \langle a_{t^*}, \widehat{\theta}_t \rangle - \varepsilon_{t+k} - 6\widetilde{\varepsilon}_{t+k} - \widetilde{\varepsilon}_t \leq \langle a_{t^*}, \widehat{\theta}_t \rangle - \varepsilon_t/2 + 4\widetilde{\varepsilon}_t + \widetilde{\varepsilon}_{t+k} \implies \\ \varepsilon_{t+k} \geq \varepsilon_t/2 - 7\widetilde{\varepsilon}_{t+k} - 5\widetilde{\varepsilon}_t \geq \varepsilon_t/2 - 12\widetilde{\varepsilon}_t. \end{aligned}$$

Since $\beta \geq 2304$, we get that $12\widetilde{\varepsilon}_t \leq \varepsilon_t/4$. Thus, we get that

$$\varepsilon_{t+k} \geq \varepsilon_t/4.$$

Notice that

$$\frac{\varepsilon_{t+k}}{\varepsilon_t} = \sqrt{\frac{q^t}{q^{t+k}}},$$

thus it immediately follows that

$$\frac{q^t}{q^{t+k}} \geq \frac{1}{16} \implies q^k \leq 16 \implies k \log q \leq \log 16 \implies k \leq 4,$$

when we pick $B = \log(T)$ batches. Thus, for every arm the bad event can happen at most 5 times, by taking a union bound over the K arms we see that the probability that our algorithm is not replicable is at most $O(K\sqrt{1/\beta})$, so picking $\beta = \Theta(K^2/\rho^2)$ suffices to get the result. □

6.8 Naive Application of Algorithm 10 with Infinite Action Space

We use a $1/T^{1/(4d+2)}$−net that has size at most $(3T)^{\frac{d}{4d+2}}$. Let $\mathcal{A}'$ be the new set of arms. We then run Algorithm 10 using $\mathcal{A}'$. This gives us the following result, that is proved right after.

Corollary 6.8.1. *Let $T \in \mathbb{N}, \rho \in (0,1]$. There is a ρ-replicable algorithm for the stochastic d-dimensional linear bandit problem with infinite arms whose expected regret is at most*

$$\mathbb{E}[R_T] \leq C \cdot \frac{T^{\frac{4d+1}{4d+2}}}{\rho^2} \sqrt{d \log(T)},$$

where $C > 0$ is an absolute numerical constant.

Proof. Since $K \leq (3T)^{\frac{d}{4d+2}}$, we have that

$$T \sup_{a \in \mathcal{A}'} \langle a, \theta^* \rangle - \mathbb{E}\left[\sum_{i=1}^{T} \langle a_t, \theta^* \rangle\right] \leq O\left(\frac{(3T)^{\frac{2d}{4d+2}}}{\rho^2}\sqrt{dT \log\left(T(3T)^{\frac{d}{4d+2}}\right)}\right) = O\left(\frac{T^{\frac{4d+1}{4d+2}}}{\rho^2}\sqrt{d \log(T)}\right)$$

Comparing to the best arm in $\mathcal{A}$, we have that:

$$T \sup_{a \in \mathcal{A}} \langle a, \theta^* \rangle - \mathbb{E}\left[\sum_{i=1}^{T} \langle a_t, \theta^* \rangle\right] = \left(T \sup_{a \in \mathcal{A}} \langle a, \theta^* \rangle - T \sup_{a \in \mathcal{A}'} \langle a, \theta^* \rangle\right) + \left(T \sup_{a \in \mathcal{A}'} \langle a, \theta^* \rangle - \mathbb{E}\left[\sum_{i=1}^{T} \langle a_t, \theta^* \rangle\right]\right)$$

Our choice of the $1/T^{1/(4d+2)}$-net implies that for every $a \in \mathcal{A}$ there exists some $a' \in \mathcal{A}'$ such that $||a - a'||_2 \leq 1/T^{1/(4d+2)}$. Thus, $\sup_{a \in \mathcal{A}} \langle a, \theta^* \rangle - \sup_{a' \in \mathcal{A}'} \langle a', \theta^* \rangle \leq ||a - a'||_2 ||\theta^*||_2 \leq 1/T^{1/(4d+2)}$. Thus, the total regret is at most

$$T \cdot 1/T^{1/(4d+2)} + O\left(\frac{T^{\frac{4d+1}{4d+2}}}{\rho^2}\sqrt{d \log(T)}\right) = O\left(\frac{T^{\frac{4d+1}{4d+2}}}{\rho^2}\sqrt{d \log(T)}\right).$$

□

6.9 The Proof of Theorem 6.4.6

Theorem. *Let $T \in \mathbb{N}, \rho \in (0, 1]$. There exists a ρ-replicable algorithm (presented in Algorithm 11) for the stochastic d-dimensional linear bandit problem with infinite action set whose expected regret is*

$$\mathbb{E}[R_T] \leq C \cdot \frac{d^4 \log(d) \log^2 \log(d) \log\log\log(d)}{\rho^2} \sqrt{T} \log^{3/2}(T),$$

for some absolute numerical constant $C > 0$, and its running time is polynomial in T^d and $1/\rho$.

Proof. First, the algorithm is ρ-replicable since in each batch we use a replicable LSE sub-routine with parameter $\rho' = \rho/B$. This implies that

$$\mathbf{Pr}[(a_1, ..., a_T) \neq (a'_1, ..., a'_T)] = \mathbf{Pr}[\exists i \in [B] : \widehat{\theta}_i \text{ was not replicable}] \leq \rho.$$

Let us fix a batch iteration $i \in [B - 1]$. Set $\mathcal{C}_i$ be the core set computed by Lemma 6.4.3. The algorithm first pulls $n_i = \frac{Cd^4 \log(d/\delta) \log^2 \log(d) \log\log\log(d)}{\varepsilon_i^2 \rho'^2}$ times each one of the arms of the i-th core set $\mathcal{C}_i$, as indicated by Lemma 6.4.5 and computes the LSE $\widehat{\theta}_i$ in a replicable way using the algorithm of Lemma 6.4.5. Let $\mathcal{E}$ be the event that over all batches the estimations are correct. We pick $\delta = 1/(2|\mathcal{A}'|T^2)$ so that this good event does hold with probability at least $1 - 1/T$. Our goal is to control the expected regret which can be written as

$$\mathbb{E}[R_T] = T \sup_{a \in \mathcal{A}} \langle a, \theta^\star \rangle - \mathbb{E} \sum_{t=1}^{T} \langle a_t, \theta^\star \rangle.$$

We have that

$$T \sup_{a \in \mathcal{A}} \langle a, \theta^\star \rangle - T \sup_{a' \in \mathcal{A}'} \langle a', \theta^\star \rangle \leq 1 \,,$$

since $\mathcal{A}'$ is a deterministic $1/T$-net of $\mathcal{A}$. Also, let us set the expected regret of the bounded action sub-problem as

$$\mathbb{E}[R'_T] = T \sup_{a' \in \mathcal{A}'} \langle a', \theta^\star \rangle - \mathbb{E} \sum_{t=1}^{T} \langle a_t, \theta^\star \rangle \,.$$

We can now employ the analysis of the finite arm case. During batch i, any active arm has gap at most $4\varepsilon_{i-1}$, so the instantaneous regret in any round is not more than $4\varepsilon_{i-1}$. The expected regret conditional on the good event $\mathcal{E}$ is upper bounded by

$$\mathbb{E}[R'_T | \mathcal{E}] \leq \sum_{i=1}^{B} 4 M_i \varepsilon_{i-1} \,,$$

where M_i is the total number of pulls in batch i (using the replicability blow-up) and ε_{i-1} is the error one would achieve by drawing q^i samples (ignoring the blow-up). Then, for some absolute constant $C > 0$, we have that

$$\mathbb{E}[R'_T | \mathcal{E}] \leq \sum_{i=1}^{B} 4 \left(q^i \frac{d^3 \log(d) \log^2 \log(d) \log\log\log(d) \log^2 T}{\rho^2} \right) \cdot \sqrt{d^2 \log(T)/q^{i-1}} \,,$$

which yields that

$$\mathbb{E}[R'_T | \mathcal{E}] \leq C \frac{d^4 \log(d) \log^2 \log(d) \log\log\log(d) \log(T) \sqrt{\log(T)}}{\rho^2} \cdot S \,,$$

where we set

$$S := \sum_{i=1}^{B} \frac{q^i}{q^{(i-1)/2}} = q^{1/2} \sum_{i=1}^{B} q^{i/2} = q^{(1+B)/2} \,.$$

We pick $B = \log(T)$ and get that, if $q = T^{1/B}$ then $S = \Theta(\sqrt{T})$. We remark that this choice of q is valid since

$$\sum_{i=1}^{B} q^i = \frac{q^{B+1} - q}{q - 1} = \Theta(q^B) - 1 \geq \frac{T \rho^2}{d^3 \log(d) \log^2 \log(d) \log\log\log(d)} \,.$$

Hence, we have that

$$\mathbb{E}[R'_T | \mathcal{E}] \leq O\left(\frac{d^4 \log(d) \log^2 \log(d) \log\log\log(d)}{\rho^2} \sqrt{T} \log^{3/2}(T) \right) \,.$$

Note that when $\mathcal{E}$ does not hold, we can bound the expected regret by $1/T \cdot T = 1$. This implies that the overall regret $\mathbb{E}[R_T] \leq 2 + \mathbb{E}[R'_T | \mathcal{E}]$ and so it satisfies the desired bound and the proof is complete. □

6.10 Deferred Lemmata

6.10.1 The Proof of Lemma 6.4.4

Proof. Consider the distribution π that is a 2-approximation to the optimal G-design and has support $|\mathcal{C}| = O(d \log\log d)$. Let $\mathcal{C}'$ be the set of arms in the support such that $\pi(a) \le c/d \log d$. We consider $\widetilde{\pi} = (1-x)\pi + xa$, where $a \in \mathcal{C}'$ and x will be specified later. Consider now the matrix $V(\widetilde{\pi})$. Using the Sherman-Morrison formula, we have that

$$V(\widetilde{\pi})^{-1} = \frac{1}{1-x} V(\pi)^{-1} - \frac{xV(\pi)^{-1}aa^\top V(\pi)^{-1}}{(1-x)^2\left(1 + \frac{1}{1-x}||a||^2_{V(\pi)^{-1}}\right)} = \frac{1}{1-x}\left(V(\pi)^{-1} - \frac{xV(\pi)^{-1}aa^\top V(\pi)^{-1}}{1-x+||a||^2_{V(\pi)^{-1}}}\right).$$

Consider any arm a'. Then,

$$||a'||^2_{V(\widetilde{\pi})^{-1}} = \frac{1}{1-x}||a||^2_{V(\pi)^{-1}} - \frac{x}{1-x} \cdot \frac{(a^\top V(\pi)^{-1}a')^2}{1-x+||a||^2_{V(\pi)^{-1}}} \le \frac{1}{1-x}||a||^2_{V(\pi)^{-1}}.$$

Note that we apply this transformation at most $O(d\log\log d)$ times. Let $\widehat{\pi}$ be the distribution we end up with. We see that

$$||a'||^2_{V(\widehat{\pi})^{-1}} \le \left(\frac{1}{1-x}\right)^{cd\log\log d} ||a||^2_{V(\pi)^{-1}} \le 2\left(\frac{1}{1-x}\right)^{cd\log\log d} d.$$

Notice that there is a constant c' such that when $x = c'/d\log d$ we have that $\left(\frac{1}{1-x}\right)^{cd\log\log d} \le 2$. Moreover, notice that the mass of every arm is at least $x(1-x)^{|\mathcal{C}|} \ge x - |\mathcal{C}|x^2 = c'/(d\log(d)) - c'' d\log\log d/(d^2\log^2(d)) \ge c/(d\log(d))$, for some absolute numerical constant $c > 0$. This concludes the claim. □

6.10.2 The Proof of Lemma 6.4.5

Proof. The proof works when we can treat $\Omega(\lceil d\log(1/\delta)\pi(a)/\varepsilon^2\rceil)$ as $\Omega(d\log(1/\delta)\pi(a)/\varepsilon^2)$, i.e., as long as $\pi(a) = \Omega(\varepsilon^2/d\log(1/\delta))$. In the regime we are in, this point is handled thanks to Lemma 6.4.4. Combining the following proof with Lemma 6.4.4, we can obtain the desired result.

We underline that we work in the fixed design setting: the arms a_i are deterministically chosen independently of the rewards r_i. Assume that the core set of Lemma 6.4.3 is the set $\mathcal{C}$. Fix the multi-set $S = \{(a_i, r_i) : i \in [M]\}$, where each arm a lies in the core set and is pulled $n_a = \Theta(\pi(a)d\log(d)\log(|\mathcal{C}|/\delta)/\varepsilon^2)$ times[3]. Hence, we have that

$$M = \sum_{a\in\mathcal{C}} n_a = \Theta\left(d\log(d)\log(|\mathcal{C}|/\delta)/\varepsilon^2\right).$$

[3] Recall that $\pi(a) \ge c/(d\log(d))$, for some constant $c > 0$, so the previous expression is $\Omega(\log(\delta/|\mathcal{C}|)/\varepsilon^2)$.

Let also $V = \sum_{i \in [M]} a_i a_i^\top$. The least-squares estimator can be written as

$$\theta_{\text{LSE}}^{(\varepsilon)} = V^{-1} \sum_{i \in [M]} a_i r_i = V^{-1} \sum_{a \in \mathcal{C}} a \sum_{i \in [n_a]} r_i(a)\,,$$

where each a lies in the core set (deterministically) and $r_i(a)$ is the i-th reward generated independently by the linear regression process $\langle \theta^\star, a \rangle + \xi$, where ξ is a fresh zero mean sub-gaussian random variable. Our goal is to reproducibly estimate the value $\sum_{i \in [n_a]} r_i(a)$ for any a. This is sufficient since two independent executions of the algorithm share the set $\mathcal{C}$ and n_a for any a. Note that the above sum is a random variable. In the following, we condition on the high-probability event that the average reward of the arm a is ε-close to the expected one, i.e., the value $\langle \theta^\star, a \rangle$. This happens with probability at least $1 - \delta/(2|\mathcal{C}|)$, given $\Omega(\pi(a) d \log(d) \log(|\mathcal{C}|/\delta)/\varepsilon^2)$ samples from arm $a \in \mathcal{C}$. In order to guarantee replicability, we will apply a result from (ILPS22). Since we will union bound over all arms in the core set and $|\mathcal{C}| = O(d \log\log(d))$ (via Lemma 6.4.3), we will make use of a $(\rho/|\mathcal{C}|)$-replicable algorithm that gives an estimate $v(a) \in \mathbb{R}$ such that

$$|\langle \theta^\star, a \rangle - v(a)| \leq \tau\,,$$

with probability at least $1 - \delta/(2|\mathcal{C}|)$. For $\delta < \rho$, the algorithm uses

$$S_a = \Omega\left(d^2 \log(d/\delta) \log^2 \log(d) \log\log\log(d)/(\rho^2 \tau^2)\right)$$

many samples from the linear regression with fixed arm $a \in \mathcal{C}$. Since we have conditioned on the randomness of $r_i(a)$ for any i, we get

$$\left| \frac{1}{n_a} \sum_{i \in [n_a]} r_i(a) - v(a) \right| \leq \left| \frac{1}{n_a} \sum_{i \in [n_a]} r_i(a) - \langle \theta^*, a \rangle \right| + |\langle \theta^*, a \rangle - v(a)| \leq \varepsilon + \tau\,,$$

with probability at least $1 - \delta/(2|\mathcal{C}|)$. Hence, by repeating this approach for all arms in the core set, we set $\theta_{\text{SQ}} = V^{-1} \sum_{a \in \mathcal{C}} a \; n_a \; v(a)$. Let us condition on the randomness of the estimate $\theta_{\text{LSE}}^{(\varepsilon)}$. We have that

$$\sup_{a' \in \mathcal{A}} |\langle a', \theta_{\text{SQ}} - \theta^\star \rangle| \leq \sup_{a' \in \mathcal{A}} |\langle a', \theta_{\text{SQ}} - \theta_{\text{LSE}}^{(\varepsilon)} \rangle| + \sup_{a' \in \mathcal{A}} |\langle a', \theta_{\text{LSE}}^{(\varepsilon)} - \theta^\star \rangle|\,.$$

Note that the second term is ε with probability at least $1 - \delta$ via Lemma 6.4.1. Our next goal is to tune the accuracy $\tau \in (0, 1)$ so that the first term yields another ε error. For the first term, we have that

$$\sup_{a' \in \mathcal{A}} |\langle a', \theta_{\text{SQ}} - \theta_{\text{LSE}}^{(\varepsilon)} \rangle| \leq \sup_{a' \in \mathcal{A}} \left| \langle a', V^{-1} \sum_{a \in \mathcal{C}} a \; n_a \; (\varepsilon + \tau) \rangle \right|$$

Note that $V = \frac{Cd\log(d)\log(|\mathcal{C}|/\delta)}{\varepsilon^2}\sum_{a\in\mathcal{C}}\pi(a)aa^\top$ and so $V^{-1} = \frac{\varepsilon^2}{Cd\log(d)\log(|\mathcal{C}|/\delta)}V(\pi)^{-1}$, for some absolute constant $C > 0$. This implies that

$$\sup_{a'\in\mathcal{A}} |\langle a', \theta_{\text{SQ}} - \theta_{\text{LSE}}^{(\varepsilon)}\rangle| \leq (\varepsilon+\tau) \sup_{a'\in\mathcal{A}} \left|\left\langle a', \frac{\varepsilon^2}{Cd\log(d)\log(|\mathcal{C}|/\delta)} V(\pi)^{-1} \sum_{a\in\mathcal{C}} \frac{Cd\log(d)\log(|\mathcal{C}|/\delta)\pi(a)}{\varepsilon^2} a\right\rangle\right| .$$

Hence, we get that

$$\sup_{a'\in\mathcal{A}} |\langle a', \theta_{\text{SQ}} - \theta_{\text{LSE}}^{(\varepsilon)}\rangle| \leq (\varepsilon+\tau) \sup_{a'\in\mathcal{A}} \left|\left\langle a', V(\pi)^{-1} \sum_{a\in\mathcal{C}} \pi(a) a\right\rangle\right| .$$

Consider a fixed arm $a' \in \mathcal{A}$. Then,

$$\begin{aligned}
\left|\left\langle a', V(\pi)^{-1} \sum_{a\in\mathcal{C}} \pi(a) a\right\rangle\right| &\leq \sum_{a\in\mathcal{C}} \pi(a) \left|\langle a', V(\pi)^{-1} a\rangle\right| \\
&\leq \sum_{a\in\mathcal{C}} \pi(a) \left(1 + \left|\langle a', V(\pi)^{-1} a\rangle\right|^2\right) \\
&= 1 + \sum_{a\in\mathcal{C}} \pi(a) \left|\langle a', V(\pi)^{-1} a\rangle\right|^2 \\
&= 1 + ||a'||^2_{V(\pi)^{-1}} \\
&\leq 4d + 1 ,
\end{aligned}$$

where the last inequality follows from the fact that π is a 4-approximation of the G-optimal design. Hence, in total, by picking $\tau = \varepsilon$, we get that

$$\sup_{a'\in\mathcal{A}} |\langle a', \theta_{\text{SQ}} - \theta^\star\rangle| \leq 11d\varepsilon .$$

Thus, for any $\varepsilon > 0$, the total number of pulls of each arm is

$$\Omega\left(d^4 \log(d/\delta) \log^2 \log(d) \log\log\log(d)/(\rho^2\varepsilon^2)\right) ,$$

to get

$$\sup_{a'\in\mathcal{A}} |\langle a', \theta_{\text{SQ}} - \theta^\star\rangle| \leq \varepsilon .$$

□

6.11 Computational Performance of Algorithm 11

In this appendix, we discuss the barriers towards computational efficiency regarding Algorithm 11. The reasons why Algorithm 11 is computationally inefficient are the following: (a) we have to compute the arm in the set of active arms that has maximum correlation with the estimate $\widehat{\theta}_i$, (b) we have to eliminate arms based on this value and (c) we have to run at each batch the Frank-Wolfe algorithm (or some other optimization method needed for Lemma 6.4.1) in order to obtain an approximate G-optimal design. As a minimal assumption in what follows, we focus on the case where the action set $\mathcal{A}$ is convex and we have access to a separation oracle for it.

Note that executing both (a) and (b) naively requires time exponential in d. However, on the one side arm elimination (issue (b)) reduces to finding the intersection of the current active set with a halfspace $\mathcal{H}$ whose normal vector is $\widehat{\theta^i}$ and the threshold is, roughly speaking, the maximum correlation. This maximum correlation can also be computed efficiently. Finding an arm with (almost) maximum correlation relates to the problem of finding a point that maximizes a linear objective under the constraint that the point lies in the intersection of the active arm set with some linear constraints. Thus, we can use the ellipsoid algorithm to implement this step.

The above discussion deals with issues (a) and (b) and, essentially, states that even with infinitely many actions, one could implement these steps efficiently. We now focus on issue (c). The Frank-Wolfe method first requires a proper initialization. As mentioned in (LS20), if the starting point is chosen to be the uniform distribution over $\mathcal{A}'$, then the number of iterations before getting a 2-approximate optimal design is roughly $\widetilde{O}(d)$. The issue is that since $\mathcal{A}'$ is exponential in d, it is not clear how to work with such an initialization efficiently. Notably there is a different initialization (Fed13; LSW20) with support $O(d)$ for which the method runs in $O(d \log\log(d))$ rounds (see Note 3 at Section 21.2 of (LS20) and (LSW20)). There are two issues: first, one requires an oracle to provide this good initialization. Second, each iteration of the Frank-Wolfe method (with current design guess π) requires computing a point in the current active set with maximum $V(\pi)^{-1}$-norm. As noted in (Tod16), a good initialization for finding a G-optimal design, i.e., a minimum volume enclosing ellipsoid (MVEE) should be sufficiently sparse (compared to the number of active arms) and assign positive mass to arms that correspond to extreme points, i.e., points that are close to the border of MVEE. The work of (KY05) provides an initial core set that depends only on d but not on the number of points. The algorithm works as follows: it runs for d iterations and, in each round, it adds 2 arms into the core set. Initially, we set the core set $\mathcal{C}_0 = \emptyset$ and let $\Psi = \{0\}$. In each iteration $i \in [d]$, the algorithm draws a random direction v_i in the orthogonal complement of Ψ (this step is replicable thanks to the shared randomness) and computes the vectors in the active arms' set with the maximum and the minimum correlation with v_i, say a_i^+, a_i^-. It then extends $\mathcal{C}_0 \leftarrow \mathcal{C}_0 \cup \{a_i^+, a_i^-\}$ and sets $\Psi \leftarrow \mathrm{span}(\Psi, \{a_i^+ - a_i^-\})$. Hence, the runtime of this algorithm corresponds to the runtime of the tasks $\max_{a \in \mathcal{A}'} \langle a, v_i \rangle$ and $\min_{a \in \mathcal{A}'} \langle a, v_i \rangle$. One can efficiently approximate these values using the ellipsoid algorithm and hence efficiently initialize the Frank-Wolfe algorithm as in (Tod16) (e.g., set the weights uniformly $1/(2d)$).

Our second challenge deals with finding a point in the active arm set with maximum $V(\pi)^{-1}$-norm for some current guess π. Even if the current active set is a polytope, finding an exact norm maximizer is NP-hard (FO85; MS86)[4]. Hence, one should focus on efficient approximation algorithms. We note that even a poly(d)-approximate maximizer is sufficient to get $\widetilde{O}(\mathrm{poly}(d)\sqrt{T})$ regret. Such an algorithm for polytopes, which gets an $1/d^2$-approximation, is provided in (Ye92; Vav93).

As a general note, if we assume that we have access to an oracle $\mathcal{O}$ that computes a 2-approximate G-optimal design in time $T_{\mathcal{O}}$, then our Algorithm 11 runs in time polynomial in $T_{\mathcal{O}}$.

[4] In fact, even finding a constant factor approximation, for some appropriate constant, is NP-hard (BR93).

Algorithm 11 Replicable LSE Algorithm for Stochastic Infinite Action Set (Theorem 6.4.6)

```
Input:  time horizon $T$, action set $\mathcal{A} \subseteq \mathbb{R}^d$, replicability $\rho$
$\mathcal{A}' \leftarrow 1/T$-net of $\mathcal{A}$
Initialization: $r \leftarrow T, B \leftarrow \log(T), q \leftarrow (T/c)^{1/B}$
for $i = 1$ to $B - 1$ do
    $q^i$ denotes the number of pulls of all arms before the replicability blow-up
    $\varepsilon_i = c \cdot d\sqrt{\log(T)/q^i}$
    The blow-up is $M_i = q^i \cdot d^3 \log(d) \log^2 \log(d) \log\log\log(d) \log^2(T)/\rho^2$
    $a_1, \dots, a_{|\mathcal{C}_i|} \leftarrow$ core set $\mathcal{C}_i$ of the design given by Lemma 6.4.3 with parameter $\mathcal{A}'$
    if $\lceil M_i \rceil > r$ then
        break
    Pull every arm $a_j$ for $N_i = \lceil M_i \rceil / |\mathcal{C}_i|$ rounds and receive rewards $r_1^{(j)}, ..., r_{N_i}^{(j)}$ for $j \in [|\mathcal{C}_i|]$
    $S_i = \{(a_j, r_t^{(j)}) : t \in [N_i], j \in [|\mathcal{C}_i|]\}$
    $\widehat{\theta}_i \leftarrow$ ReplicableLSE$(S_i, \rho' = \rho/(dB), \delta = 1/(2|\mathcal{A}'|T^2), \tau = \min\{\varepsilon_i, 1\})$
    $r \leftarrow r - \lceil M_i \rceil$
    for $a \in \mathcal{A}'$ do
        if $\langle a, \widehat{\theta}_i \rangle < \max_{a \in \mathcal{A}'} \langle a, \widehat{\theta}_i \rangle - 2\varepsilon_i$ then
            Remove $a$ from $\mathcal{A}'$
In the last batch play $\arg\max_{a \in \mathcal{A}'} \langle a, \widehat{\theta}_{B-1} \rangle$

ReplicableLSE$(S, \rho, \delta, \tau)$
for $a \in \mathcal{C}$ do
    $v(a) \leftarrow$ ReplicableSQ$(\phi : x \in \mathbb{R} \mapsto x \in \mathbb{R}, S, \rho, \delta, \tau)$        ▷ (ILPS22)
return $(\sum_{j \in |S|} a_j a_j^\top)^{-1} \cdot (\sum_{a \in \mathcal{C}} a \; n_a \; v(a))$
```

Chapter 7

Statistical Indistinguishability of Learning Algorithms

7.1 Replicability, Differential Privacy and TV Indistinguishability

We shortly remind the reader the three crucial definitions of this chapter, which had been extensively discussed in Section 2.4.5

Definition 7.1.1 (Replicability (ILPS22))**.** *Let $\mathcal{R}$ be a distribution over random strings. A learning algorithm A is n-sample ρ-replicable if for any distribution $\mathcal{D}$ over inputs and two independent sets $S, S' \sim \mathcal{D}^n$ it holds that*

$$\mathop{\mathbf{Pr}}_{S,S' \sim \mathcal{D}^n, r \sim \mathcal{R}}[A(S, r) \neq A(S', r)] \leq \rho\,.$$

Definition 7.1.2 (Approximate Differential Privacy (DKM$^+$06))**.** *A learning rule A is an n-sample (ϵ, δ)-differentially private if for any pair of samples $S, S' \in (\mathcal{X} \times \{0, 1\})^n$ that disagree on a single example, the induced posterior distributions $A(S)$ and $A(S')$ are (ϵ, δ)-indistinguishable.*

Definition 7.1.3 (Total Variation Indistinguishability)**.** *A learning rule A is n-sample ρ-*TV *indistinguishable if for any distribution over inputs $\mathcal{D}$ and two independent sets $S, S' \sim \mathcal{D}^n$ it holds that*

$$\mathop{\mathbb{E}}_{S,S' \sim \mathcal{D}^n}[d_{\mathrm{TV}}(A(S), A(S'))] \leq \rho\,.$$

In this chapter, we investigate the connections between TV indistinguishability, replicability and differential privacy.

7.2 Generalization Bounds for TV Indistinguishable Learners

As a warmup, we start by proving a generalization result for TV indistinguishable learners. Recall that if we *fix* some binary classifier we can show, using standard concentration bounds, that its performance on a sample is close to its performance on the underlying population. However, when we train an ML algorithm using a dataset S to output a classifier h we cannot just use the fact that it has small loss on S to claim that its loss on the population is small because h depends on S. The following result shows that we can get such generalization bounds if A is a ρ-TV indistinguishable algorithm. We remark that a similar result regarding replicable algorithms appears in (ILPS22). The formal proof, stated in a slightly more general way, is in Section 7.12.

Proposition 7.2.1 (TV Indistinguishability Implies Generalization)**.** *Let $\delta, \rho \in (0,1)^2$. Let $\mathcal{D}$ be a distribution over inputs and $S = \{(x_i, y_i)\}_{i\in[n]}$ be a sample of size n drawn i.i.d. from $\mathcal{D}$. Let $h : \mathcal{X} \to \{0,1\}$ be the output of an n-sample ρ-*TV *indistinguishable learning rule A with input S. Then, with probability at least $1-\delta-4\sqrt{\rho}$ over S, it holds that,*

$$\left| \mathop{\mathbb{E}}_{h\sim A(S)} [L(h)] - \mathop{\mathbb{E}}_{h\sim A(S)} \left[\widehat{L}(h)\right] \right| \leq \sqrt{\frac{\log(2/\delta)}{2n}} + \sqrt{\rho}\,,$$

where $L(h) \triangleq \mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$ and $\widehat{L}(h) \triangleq \frac{1}{n}\sum_{(x,y)\in S} 1\{h(x) \neq y\}$.

7.3 Related Work

This chapter falls in the research agenda of replicable algorithm design, which was initiated by (ILPS22). In particular, (ILPS22) introduced the notion of replicable learning algorithms, established that any statistical query algorithm can be made replicable, and designed replicable algorithms for various applications such as halfspace learning. Next, (AJJ+22) studied reproducibility in optimization and (EKK+22) provided replicable bandit algorithms.

The most closely related prior work to ours is the recent paper by (BGH+23). In particular, as we discuss below in greater detail, an alternative proof of the equivalence between TV indistinguishability, replicability, and differential privacy follows from (BGH+23). In contrast with our equivalence, the transformations by (BGH+23) are restricted to finite classes. On the other hand, (BGH+23) give a constructive proof whereas our proof is purely information-theoretic.

In more detail, (BGH+23) establish a variety of equivalences between different notions of stability such as differential privacy, replicability, and one-way perfect generalization, and the latter contains TV indistinguishability as a special case:

Definition 7.3.1 ((One-Way) Perfect Generalization (CLN$^+$16; BF16))**.** *A learning rule $A : \mathcal{X}^n \to \mathcal{Y}$ is $(\beta, \varepsilon, \delta)$-perfectly generalizing if, for every distribution $\mathcal{D}$ over $\mathcal{X}$, there exists a distribution $\mathcal{P}_{\mathcal{D}}$ such that, with probability at least $1-\beta$ over S consisting of n i.i.d. samples from $\mathcal{D}$, and every set of outcomes $\mathcal{O} \subseteq \mathcal{Y}$*

$$e^{-\varepsilon}\left(\Pr_{\mathcal{P}_{\mathcal{D}}}[\mathcal{O}] - \delta\right) \leq \Pr[A(S) \in \mathcal{O}] \leq e^{\epsilon} \Pr_{\mathcal{P}_{\mathcal{D}}}[\mathcal{O}] + \delta\,.$$

Moreover, A is $(\beta, \epsilon, \delta)$-one-way perfectly generalizing if $\Pr[A(S) \in \mathcal{O}] \leq e^{\epsilon} \Pr_{\mathcal{P}_{\mathcal{D}}}[\mathcal{O}] + \delta$.

Note indeed that plugging $\epsilon = 0$ to the definition of perfect generalization specializes the above definition to an equivalent variant of TV indistinguishability (see also Definition 7.7.9). (BGH$^+$23) derives an equivalence between replicability and one-way perfect generalization with $\epsilon > 0$. However, in a personal communication they pointed out to us that their argument also applies to the case $\epsilon = 0$, and hence to TV indistinguishability. In more detail, an intermediate step of their proof shows that any $(\beta, \epsilon, \delta)$-perfectly generalizing algorithm A is also $(\beta, 0, 2\varepsilon + \delta)$-perfectly generalizing, which is qualitatively equivalent with our main definition (see Definition 7.1.3). As noted earlier our proof applies more generally to infinite countable domains but is non-constructive.

Differential Privacy. Differential privacy (Dwo08; DRV10; Vad17; DR14) is quite closely related to replicability. The first connection between replicability and DP in the context of PAC learning was, implicitly, established by (GKM21) (for finite domains $\mathcal{X}$), via the technique of correlated sampling (see Section 7.7.4) and the notion of pseudo-global stability (which is equivalent to replicability as noticed by (ILPS22)):

Definition 7.3.2 (Pseudo-Global Stability (GKM21))**.** *Let $\mathcal{R}$ be a distribution over random strings. A learning algorithm A is said to be n-sample (η, ν)-pseudo-globally stable if for any distribution $\mathcal{D}$ there exists a hypothesis h_r for every $r \in \operatorname{supp}(\mathcal{R})$ (depending on $\mathcal{D}$) such that*

$$\Pr_{r \sim \mathcal{R}}\left[\Pr_{S \sim \mathcal{D}^n}[A(S, r) = h_r] \geq \eta\right] \geq \nu\,.$$

The high-level connection between these notions appears to boil down to the notion of stability (BE02; PRMN04; DFH$^+$15; ALMT17; BNS$^+$16a; LM20) (see (ABL$^+$22) for further details between stability, online learnability and differential privacy). In particular, (GGKM21) showed that a class of finite Littlestone dimension admits a list-globally stable learner (see Theorem 18 in (GKM21)). The work of (GKM21) (among other things) showed (i) how to perform a reduction from list-global stability to pseudo-global stability via correlated sampling in finite domains (see Theorem 20 in (GKM21)) and (ii) how to perform a reduction from pseudo-global stability to approximate DP via DP selection (see Theorem 25 in (GKM21)). We highlight that this equivalence between differential privacy and replicability for finite domains was made formal by (BGH$^+$23) and was extended to arbitrary statistical tasks.

TV Stability. The definition of TV indistinguishability that we propose has close connections with the definition of TV stability. This notion has appeared in the context of adaptive data analysis. The work of (BNS$^+$16a) studied the following problem: suppose there is an unknown distribution P and a set S of n independent samples drawn i.i.d. from P. The goal is to design an algorithm that, with input S, will accurately answer a sequence of adaptively chosen queries about the unknown distribution P. The main question is how many samples must one draw from the distribution, as a function of the type of queries, the number of queries, and the desired level of accuracy to perform well? (BNS$^+$16a) provide various results that rely on the connections between algorithmic stability, differential privacy and generalization. To this end, they think of differential privacy as max-KL stability and study the performance of other notions of stability such as TV stability. Crucially, in their definition, TV stability considers any pair of neighboring datasets S, S' and not two independent draws from P. More concretely, they propose the following definition.

Definition 7.3.3 (Total Variation Stability (BNS$^+$16a)). *A learning rule A is n-sample ρ-*TV *stable if for any pair of samples $S, S' \in (\mathcal{X} \times \{0,1\})^n$ that disagree on a single example, it holds that $d_{\mathrm{TV}}(A(S), A(S')) \leq \rho$.*

We underline that for any constant ρ[1] it is not challenging to obtain a ρ-TV stable algorithm in the learning setting we are interested in. It suffices to just sub-sample a small enough subset of the data. Hence, any class with finite VC dimension is TV stably learnable under this definition. As it is evident from our results (cf. Theorem 7.5.2), this is in stark contrast with the definition we propose. We remind the readers that just sub-sampling the dataset is not enough to achieve differential privacy. This is because it is required that $\delta = o(1/n)$. We remark that the definition of total variation stability à la (BNS$^+$16a) also appears in (RRT$^+$16).

The above definition of TV stability has close connections to machine unlearning. This problem refers to the ability of a user to delete their data that were used to train a ML algorithm. When this happens, the machine learning algorithm has to move to a state as if it had never used that data for training, hence the term *machine unlearning*. One can see that Definition 7.3.3 is suitable for this setting since it states that if one point of the dataset is deleted, the distribution of the algorithm should not be affected very much. For convex risk minimization problems, (UMR$^+$21) design TV stable algorithms based on noisy Stochastic Gradient Descent (SGD). Such approaches lead to the design of efficient unlearning algorithms,

[1] In fact, even for $\rho \geq 1/n^c, 0 < c < 1$.

which are based on sub-sampling the dataset and constructing a maximal coupling of Markov chains for the noisy SGD procedure.

KL Stability and PAC-Bayes. In Section 7.7.3 we provide some equivalent definitions to TV indistinguishability. In particular, Definition 7.7.9 has connections with the line of work that studies distribution-dependent generalization bounds. To be more precise, if instead of the TV distance we use the KL divergence to measure the distance between the prior and the output of the algorithm we get the definition of the quantity that is used to derive PAC-Bayes generalization bounds. Interestingly, (LM20) show that the PAC-Bayes framework cannot be used to derive distribution-free PAC learning bounds for classes that have infinite Littlestone dimension; they show that for any algorithm that learns 1-dimensional linear classifiers (thresholds), there exists a realizable distribution for which PAC-Bayes bounds are trivial. Recently, a similar PAC-Bayes framework was proposed in (AEMM22), where the KL divergence is replaced with a general family of Integral Probability Metrics (cf. Definition 7.7.2).

Probably Eventually Correct Learning. The work of (MM22) introduced the *Probably Eventually Correct* (PEC) model of learning. In this model, a learner outputs the same hypothesis[2], with probability one, after a uniformly bounded number of revisions. Intuitively, this corresponds to the property that the global stability parameter is close to 1. Interestingly, prior work on global stability (BLM20; GGKM21) had characterized *Littlestone classes* as being PAC learnable by an algorithm which outputs some fixed hypothesis with nonzero probability. However, the frequency of this hypothesis was typically very small and its loss was a priori non-zero. (MM22) give a new characterization to Littlestone classes by identifying them with the classes that can be PEC learned in a stable fashion. Informally, this means that the learning rule for $\mathcal{H}$ stabilizes on some hypothesis after changing its mind at most L times, where L is the Littlestone dimension of $\mathcal{H}$ (cf. Definition 7.7.3). Interestingly, (MM22) manage to show that the well-known *Standard Optimal Algorithm* (SOA) (Lit88) is a stable PEC learner, using tools from the theory of universal learning (BHM$^+$21; BHM$^+$22; KVK22; HKMV22). Moreover, they list various different notions of algorithmic stability and show that they all have something in common: a class $\mathcal{H}$ is learnable by such learners if and only if its Littlestone dimension is finite. Our main result shows that, indeed, classes that are learnable by TV indistinguishable learners fall into that category.

[2] Except maybe for a subset of $\mathcal{X}$ that has measure zero under the data-generating distribution.

7.4 TV Indistinguishability and Replicability

Our information-theoretic definition of TV indistinguishability seems to put weaker restrictions on learning rules than the notion of replicability in two ways: (i) it allows for *arbitrary* couplings between the two executions of the algorithm (recall the coupling definition of TV distance, see Eq.(2.1)), and, (ii) it allows for *different* couplings between every pair of datasets S, S' (the optimal coupling in the definition of TV distance will depend on S, S' of Definition 7.1.3). In short, our definition allows for *arbitrary data-dependent* couplings, instead of just sharing the randomness across two executions. TV indistinguishability can be viewed as a statistical generalization of replicability (cf. Definition 7.1.1) since it describes a property of *learning rules* rather than *learning algorithms.*

In this section, we will show that TV indistinguishability and replicability are (perhaps surprisingly) equivalent in a rather strong sense: under a mild measure-theoretic condition, every TV indistinguishable algorithm can be converted into an *equivalent* replicable one by *re-interpreting* its internal randomness. This will be made formal shortly.

We start by showing that any replicable algorithm is TV indistinguishable.

Theorem 7.4.1 (Replicability ⇒ TV Indistinguishability)**.** *If a learning rule A is n-sample ρ-replicable, then it is also n-sample ρ-*TV *indistinguishable.*

Proof. Fix some distribution $\mathcal{D}$ over inputs. Let A be n-sample ρ-replicable with respect to $\mathcal{D}$. For the random variables $A(S), A(S')$ where $S, S' \sim \mathcal{D}^n$ are two independent samples and using Eq.(2.1), we have

$$\mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S), A(S'))] = \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}\left[\inf_{(h,h')\sim\Pi(A(S),A(S'))} \mathbf{Pr}[h \neq h']\right]. \tag{7.1}$$

Let $\mathcal{R}$ be the source of randomness that A uses. The expected optimal coupling of Eq.(7.1) is at most $\mathbb{E}_{S,S'\sim\mathcal{D}^n}\left[\mathbf{Pr}_{r\sim\mathcal{R}}[A(S,r) \neq A(S',r)]\right]$. This inequality follows from the fact that using shared randomness between the two executions of A is a particular way to couple the two random variables. To complete the proof, it suffices to notice that this upper bound is equal to

$$\mathop{\mathbf{Pr}}_{S,S'\sim\mathcal{D}^n, r\sim\mathcal{R}}[A(S,r) \neq A(S',r)] \leq \rho\,.$$

The last inequality follows since A is ρ-replicable. □

We now deal with the opposite direction, i.e., we show that TV indistinguishability implies replicability. In order to be formal, we need to discuss some measure theoretic properties first. Let us recall the definition of absolute continuity for two measures.

Definition 7.4.2 (Absolute Continuity)**.** *Consider two measures P, Q on a σ-algebra $\mathcal{B}$ of subsets of Ω. We say that P is absolutely continuous with respect to Q if for any $E \in \mathcal{B}$ such that $Q(E) = 0$, it holds that $P(E) = 0$.*

Since the learning rules induce posterior distributions over hypotheses, this definition extends naturally to such rules.

Definition 7.4.3. *Given learning rule A, distribution over inputs $\mathcal{D}$ and reference probability measure $\mathcal{P}$, we say that A is absolutely continuous with respect to $\mathcal{P}$ on inputs from $\mathcal{D}$ if, for almost every sample S drawn from $\mathcal{D}$, the posterior distribution $A(S)$ is absolutely continuous with respect to $\mathcal{P}$.*

In the previous definition, we fixed the data-generating distribution $\mathcal{D}$. We next consider its distribution-free version.

Definition 7.4.4. *Given learning rule A and reference probability measure $\mathcal{P}$, we say that A is absolutely continuous with respect to $\mathcal{P}$ if, for any distribution over inputs $\mathcal{D}$, A is absolutely continuous with respect to $\mathcal{P}$ on inputs from $\mathcal{D}$.*

If $\mathcal{X}$ is finite, then one can take $\mathcal{P}$ to be the uniform probability measure over $\{0,1\}^{\mathcal{X}}$ and any learning rule is absolutely continuous with respect to $\mathcal{P}$. We now show how we can find such a prior $\mathcal{P}$ in the case where $\mathcal{X}$ is countable.

Claim 19 (Reference Probability Measure for Countable Domains)**.** *Let $\mathcal{X}$ be a countable domain and A be a learning rule. Then, there is a reference probability measure $\mathcal{P}$ such that A is absolutely continuous with respect to $\mathcal{P}$.*

Proof. Since $\mathcal{X}$ is countable, for a fixed n, we can consider an enumeration of all the n-tuples $\{S_i\}_{i\in\mathbb{N}}$. Then, we can take $\mathcal{P}$ to be a countable mixture of these probability measures, i.e., $\mathcal{P} = \sum_{i=1}^{\infty} \frac{1}{2^i} A(S_i)$. Notice that since, each $A(S_i)$ is a measure and $1/2^i > 0$ for $i \in \mathbb{N}$, and, $\sum_{i=1}^{\infty} 1/2^i = 1$, we have that $\mathcal{P}$ is indeed a probability measure. We now argue that each $A(S_i)$ is absolutely continuous with respect to $\mathcal{P}$. Assume towards contradiction that this is not the case and let $E \in \mathcal{B}$ be a set such that $\mathcal{P}(E) = 0$ but $A(S_j)(E) \neq 0$, for some $j \in \mathbb{N}$. Notice that $A(S_j)$ appears with coefficient $1/2^j > 0$ in the mixture that we consider, hence if $A(S_j)(E) > 0 \implies 1/2^j A(S_j)(E) > 0$. Moreover $A(S_i)(E) \geq 0, \forall i \in \mathbb{N}$, which means that $\mathcal{P}(E) > 0$, so we get a contradiction. □

We next define when two learning rules A, A' are equivalent.

Definition 7.4.5 (Equivalent Learning Rules)**.** *Two learning rules A, A' are equivalent if for every sample S it holds that $A(S) = A'(S)$, i.e., for the same input they induce the same distribution over hypotheses.*

In the next result, we show that for every TV indistinguishable algorithm A, that is absolutely continuous with respect to some reference probability measure $\mathcal{P}$, there exists an equivalent learning rule which is replicable.

Theorem 7.4.6 (TV Indistinguishability ⇒ Replicability)**.** *Let $\mathcal{P}$ be a reference probability measure over $\{0,1\}^{\mathcal{X}}$, and let A be a learning rule that is n-sample ρ-*TV *indistinguishable and absolutely continuous with respect to $\mathcal{P}$. Then, there exists an equivalent learning rule A' that is n-sample $\frac{2\rho}{1+\rho}$-replicable.*

In this section, we only provide a sketch of the proof and we refer the reader to Section 7.9.1 for the complete one. Let us first state how we can use the previous result when $\mathcal{X}$ is countable.

Corollary 7.4.7. *Let $\mathcal{X}$ be a countable domain and let A be a learning rule that is n-sample ρ-*TV *indistinguishable. Then, there exists an equivalent learning rule A' that is n-sample $\frac{2\rho}{1+\rho}$-replicable.*

The proof of this result follows immediately from Claim 19 and Theorem 7.4.6.

Proof Sketch of Theorem 7.4.6. Let us consider a learning rule A satisfying the conditions of Theorem 7.4.6. Fix a distribution $\mathcal{D}$ over inputs. The crux of the proof is that given two random variables X, Y whose TV distance is bounded by ρ, we can couple them using only a carefully designed source of shared randomness $\mathcal{R}$ so that the probability that the realizations of these random variables differ is at most $2\rho/(1+\rho)$. We can instantiate this observation with $X = A(S)$ and $Y = A(S')$. Crucially, in the countable $\mathcal{X}$ setting, we can pick the shared randomness $\mathcal{R}$ in a way that only depends on the learning rule A, but not on S or S'. Let us now describe how this coupling works. Essentially, it can be thought of as a generalization of the von Neumann rejection-based sampling which does not necessarily require that the distribution has bounded density. Following (AS19), we pick $\mathcal{R}$ to be a Poisson point process which generates points of the form (h, y, t) with intensity[3] $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, where $\mathcal{P}$ is a reference probability measure with respect to which A is absolutely continuous and Leb is the Lebesgue measure over $\mathbb{R}_+$. Intuitively, $h \sim \mathcal{P}$ lies in the hypotheses' space, y is a non-negative real value and t corresponds to a time value. The coupling mechanism performs *rejection sampling* for each distribution we would like to couple (here $A(S)$ and $A(S')$): it checks (in the ordering indicated by the time parameter) for each point (h, y, t) whether $f(h) > y$ (i.e., if y falls below the density curve f at h) and accepts the first point that satisfies this condition. In the formal proof, there will be two density functions; f (resp. f') for the density function of $A(S)$ (resp. $A(S')$). We also refer to Figure 7.1. One can show (see Theorem 7.7.13) that $\mathcal{R}$ gives rise to a coupling between $A(S)$ and $A(S')$ under the condition that both measures are absolutely continuous with respect to the reference probability measure $\mathcal{P}$.

This coupling technique appears in (AS19). We can then apply it and get

$$\mathop{\mathbf{Pr}}_{r\sim\mathcal{R}}[A(S,r) \neq A(S',r)] \leq \frac{2d_{\mathrm{TV}}(A(S), A(S'))}{1 + d_{\mathrm{TV}}(A(S), A(S'))} .$$

Taking the expectation with respect to the draws of S, S', we show (after some algebraic manipulations) that $\mathbf{Pr}_{S,S'\sim\mathcal{D}^n r\sim\mathcal{R}}[A(S,r) \neq A(S',r)] \leq 2\rho/(1+\rho)$. We conclude this section with the following remarks.

[3] Roughly speaking, a point process is a (general) Poisson point process with intensity λ if (i) the number of points in a bounded Borel set E is a Poisson random variable with mean $\lambda(E)$ and (ii) the numbers of points in n disjoint Borel sets forms n independent random variables. For further details, we refer to (LP17a).

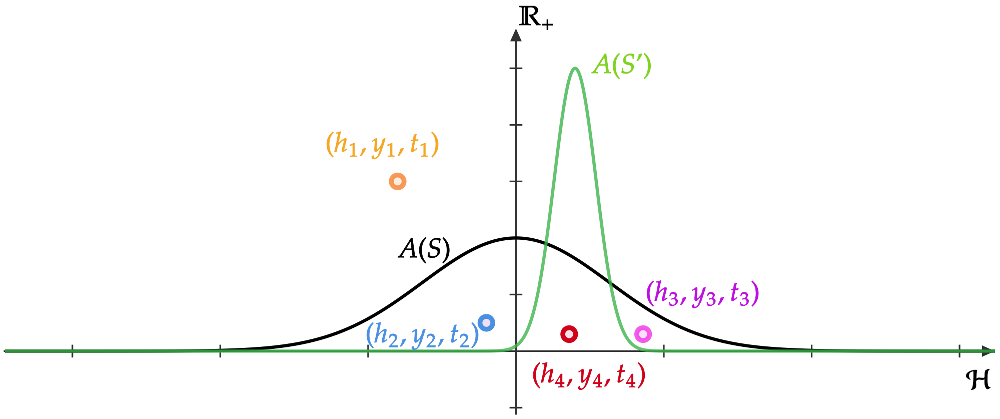


Figure 7.1: Our goal is to couple $A(S)$ with $A(S')$, where these two distributions are absolutely continuous with respect to the reference probability measure $\mathcal{P}$. A sequence of points of the form (h, y, t) is generated by the Poisson point process with intensity $\mathcal{P} \times \text{Leb} \times \text{Leb}$ where $h \sim \mathcal{P}$, $(y, t) \in \mathbb{R}_+^2$ and Leb is the Lebesgue measure over $\mathbb{R}_+$ (note that we do not have upper bounds for the densities). Intuitively, h lies in the hypotheses' space, y is a non-negative real value and t corresponds to a time value. Let f be the Radon-Nikodym derivate of $A(S)$ with respect to $\mathcal{P}$. We assign the first (the one with minimum t) value h to $A(S)$ that satisfies the property that $f(h) > y$, i.e., y falls below the density curve of $A(S)$. We assign a hypothesis to $A(S')$ in a similar manner. This procedure defines a data-independent way to couple the two random variables and naturally extends to multiple ones. In the figure's example, we set $A(S) = h_2$ and $A(S') = h_4$ given that $t_1 < t_2 < t_3 < t_4$.

Remark 7 (General Equivalence)**.** *In Section 7.9.2, we discuss how the above equivalence actually holds for general statistical tasks beyond binary classification. We first generalize the notions of indistinguishability, replicability and* TV *indistinguishability for general input spaces $\mathcal{I}$ and output spaces $\mathcal{O}$. We then discuss that replicability and* TV *indistinguishability remain equivalent (under the same measure theoretic conditions) in these more general abstract learning scenarios.*

Remark 8 (Implementation of the Coupling)**.** *We note that, in order to implement algorithm A' of Theorem 7.4.6, we need sample access to a Poisson point process with intensity $\mathcal{P} \times \text{Leb} \times \text{Leb}$, where $\mathcal{P}$ is the reference probability measure from Claim 19 and* Leb *is the Lebesgue measure over $\mathbb{R}_+$. Importantly, $\mathcal{P}$ depends only on A. Moreover, we need full access to the values of the density f_i of the distribution $A(S_i)$ with respect to the reference probability measure $\mathcal{P}$, for any sample S_i. We underline that these quantities do not depend on the data-generating distribution $\mathcal{D}$ (since we iterate over any possible sample).*

Remark 9 (TV Indistinguishability vs. Replicability)**.** *Notice that in the definition of replicability (cf. Definition 7.1.1) the source of randomness $\mathcal{R}$ needs to be specified and by changing it we can observe different behaviors for coupled executions of the algorithm. On the other hand, the definition of* TV *indistinguishability (cf. Definition 7.1.3) does not require the specification of $\mathcal{R}$ as it states a property of the posterior distribution of the learning rule.*

7.5 TV Indistinguishability and Differential Privacy

In this section we investigate the connections between TV indistinguishability and approximate DP in binary classification. Consider a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. We will say that $\mathcal{H}$ is learnable by a ρ-TV indistinguishable learning rule A if this rule satisfies the notion of learnability under the standard realizable PAC learning model and is ρ-TV indistinguishable (see Definition 7.7.5).

The main result of this section is an equivalence between approximate DP and TV indistinguishability for countable domains $\mathcal{X}$, in the context of PAC learning. We remark that the equivalence of differential privacy with the notion of replicability is formally stated for finite outcome spaces (i.e., under the assumption that $\mathcal{X}$ is finite) due to the use of a specific correlated sampling strategy for the direction that "DP implies replicability" in the context of classification (GKM21). Moreover, (BGH$^+$23) gave a constructive way to transform a DP algorithm to a replicable one for general statistical tasks and for finite domains. Thus, combining our results in Section 7.4 and the result of (GKM21; ILPS22; BGH$^+$23), the equivalence of TV indistinguishability and DP for *finite* domains is immediate. We will elaborate more on the differences of our approach and (GKM21; BGH$^+$23) later on. We also discuss our coupling and correlated sampling in Section 7.7.4.

Recall that a learner is (α, β)-accurate if its misclassification probability is at most α with probability at least $1 - \beta$.

Theorem 7.5.1 ((ϵ, δ)-DP $\Rightarrow$ TV Indistinguishability)**.** *Let $\mathcal{X}$ be a (possibly infinite) domain and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. Let $\gamma \in (0, 1/2), \alpha, \beta, \rho \in (0,1)^3$. Assume that $\mathcal{H}$ is learnable by an n-sample $(1/2 - \gamma, 1/2 - \gamma)$-accurate $(0.1, 1/(n^2 \log(n)))$-differentially private learner. Then, it is also learnable by an (α, β)-accurate ρ-*TV *indistinguishable learning rule.*

Proof Sketch of Theorem 7.5.1. The proof goes through the notion of global stability (cf. Definition 7.7.8). The existence of an (ϵ, δ)-DP learner implies that the hypothesis class $\mathcal{H}$ has finite Littlestone dimension (ALMM19) (cf. Theorem 7.10.3). Thus, we know that there exists a ρ-globally stable learner for $\mathcal{H}$ (BLM20) (cf. Theorem 7.10.4). The next step is to use the replicable heavy-hitters algorithm (cf. Algorithm 12, (ILPS22)) with frequency parameter $O(\rho)$ and replicability parameter $O(\rho')$, where $\rho' \in (0,1)$ is the desired TV indistinguishability parameter of the learning rule. The global stability property implies that the list of heavy-hitters will be non-empty and it will contain at least one hypothesis with small error rate, with high probability. Finally, since the list of heavy-hitters is finite and has bounded size, we feed the output into the replicable agnostic learner (cf. Algorithm 13). Thus, we have designed a replicable learner for $\mathcal{H}$, and Theorem 7.4.1 shows that this learner is also TV indistinguishable.

The formal proof of Theorem 7.5.1 is deferred to Section 7.10.2. We also include a result which shows that *list-global* stability implies TV indistinguishability for general domains and general statistical tasks, which could be of independent interest (cf. Proposition 7.10.12).

We proceed to the opposite direction where we provide an algorithm that takes as input a TV indistinguishable learning rule for $\mathcal{H}$ and outputs a learner for $\mathcal{H}$ which is (ϵ, δ)-DP. In this direction countability of $\mathcal{X}$ is crucial.

Theorem 7.5.2 (TV Indistinguishability $\Rightarrow$ (ϵ, δ)-DP)**.** *Let $\mathcal{X}$ be a countable domain. Assume that $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is learnable by an (α, β)-accurate ρ-TV indistinguishable learner A, for some $\rho \in (0,1), \alpha \in (0,1/2), \beta \in \left(0, \frac{1-\rho}{1+\rho}\right)$. Then, for any $(\alpha', \beta', \varepsilon, \delta) \in (0,1)^4$, it is also learnable by an $(\alpha + \alpha', \beta')$-accurate (ε, δ)-differentially private learner A'.*

We refer to Section 7.10.4 for the proof. In the above statements, we omit the details about the sample complexity. We refer to Proposition 7.10.12 and Proposition 7.10.16 for these details. Let us now comment on the differences between (GKM21; BGH$^+$23) which establish a transformation from a replicable learner to an approximately DP learner and our result. The high-level idea to obtain both of these results is similar. Essentially, the proof of (GKM21; BGH$^+$23) can be viewed as a coupling between sufficiently many posteriors of the replicable learning rule using *shared randomness* in order to achieve this coupling. In our proof, instead of using shared randomness we use the reference measure we described in previous sections to achieve this coupling. We remark that we could have obtained the same qualitative result, i.e., that TV indistinguishability implies approximate DP, by using the transformation from replicability to approximate DP of (GGKM21; BGH$^+$23) in a black-box manner along with our result that TV indistinguishability implies replicability (cf. Theorem 7.4.6). However, this leads to worse guarantees in terms of the range of the parameters $\alpha, \beta, \delta, \epsilon, \rho$ than the ones stated in Theorem 7.5.2. Thus, we have chosen to do a more careful analysis based on the coupling we proposed that leads to a stronger quantitative result. More concretely, the proof in (GKM21; BGH$^+$23) starts by sampling many random strings independently of the dataset $\{S_i\}_{i\in[k]}$ and considers many executions of the algorithm using the same random strings but different data. In our algorithm we first sample the sets $\{S_i\}_{i\in[k]}$ and then we consider an optimal coupling along the $\{A(S_i)\}_{i\in[k]}$ which is also independent of the dataset, thus it satisfies the DP requirements. Moreover, our procedure covers a wider range of parameters α, β, ρ

compared to (GKM21). The reason we need countability of $\mathcal{X}$ is because it allows us to design a *data-independent* reference probability measure $\mathcal{P}$, the same one as in Claim 19. Then, using this reference probability measure for the coupling helps us establish the DP properties. Nevertheless, we propose a simple change to our approach which we conjecture applies to general domains $\mathcal{X}$ and we leave it open as an interesting future direction. For a more detailed discussion, we refer the reader to Section 7.10.5.

Interestingly, we underline that, as is shown in (GKM21; BGH$^+$23) and as opposed to Theorem 7.5.2, replicability implies DP in general spaces (cf. Theorem 7.10.5).

We conclude this section by stating a general equivalence between (ϵ, δ)-DP and replicability for PAC learning, that follows from the previous discussion, in particular by combining Theorem 7.10.5 (GKM21; BGH$^+$23), and Lemma 7.10.8.

Theorem 7.5.3 (Replicability $\iff$ Differential Privacy in PAC Learning)**.** *Let $\mathcal{X}$ be a (possibly infinite) domain and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. Then, $\mathcal{H}$ is replicably learnable if and only if it is approximately-DP learnable.*

Remark 10 (Dependence on the Parameters)**.** *In the case of* TV *indistinguishability $\Rightarrow$* DP, *the blowup in the sample complexity is stated explicitly in Proposition 7.10.16.*

For the direction DP $\Rightarrow$ TV *indistinguishability it is a bit trickier to state the exact sample complexity blow-up because we do not make explicit use of the* DP *learner. Instead, we use the fact that the existence of a non-trivial* DP *learner implies that the class has finite Littlestone dimension and then we use an appropriate algorithm that is known to work for such classes. In this case, it suffices to let the parameters of the* DP *learner to be $\epsilon \in (0, 0.1), \delta \in \left(0, \frac{1}{n^2 \log(n)}\right), \alpha \in (0, 1/2), \beta \in (0, 1/2)$ and the parameters of the desired* TV *indistinguishable (α', β')-accurate learner are unconstrained, i.e., $\rho \in (0,1), \alpha' \in (0,1), \beta' \in (0,1)$. If we denote the Littlestone dimension of the class by L, then, as shown in Proposition 7.10.12 the sample complexity of the* TV *indistinguishable learner is* $\mathrm{poly}(L, 1/\rho, 1/\alpha', \log(1/\beta'))$[4].

Remark 11 (Beyond Binary Classification)**.** *The only transformation that is restricted to binary classification is the one from* DP *to* TV *indistinguishability. All the other transformations, (and the boosting algorithms that we present in the upcoming section), extend to general statistical tasks. Let us now shortly discuss how to extend our result e.g., to the multi-class setting, using results from the private multiclass learning literature (JKT20; SBG21). (JKT20) showed that private multiclass learnability implies finite multiclass Littlestone dimension and (SBG21) showed how to extend the binary list-globally stable learner that we use to the multiclass setting. Using these two main ingredients, the rest of our approach for the*

[4] This holds under the (standard) assumption that uniform convergence holds for Littlestone classes. If this is not the case, we get $\mathrm{poly}(2^{2^L}, 1/\rho, 1/\alpha', \log(1/\beta'))$ sample complexity (Corollary 7.10.9).

binary classification setting should extend to the multiclass setting. The extension to the regression problem seems to be more challenging. Even though (JKT20) showed that private regression implies finiteness of some appropriate Littlestone dimension, it is not clear yet how to derive a (list-)globally stable algorithm for this problem.

7.6 Amplifying and Boosting TV Indistinguishable Algorithms

In this section we study the following fundamental question.

Question 3. *Consider a weak* TV *indistinguishable learning rule both in terms of the indistinguishability parameter and the accuracy. Is it possible to amplify its indistinguishability and to boost its accuracy?*

For instance, in the context of approximate differential privacy, a series of works has lead to (constructive) algorithms that boost the accuracy and amplify the privacy guarantees (e.g., (DRV10; BLM20; BGH$^+$23)). This result builds upon the equivalence of online learnability and approximate differential privacy. Our result relating DP to TV indistinguishability implies the following existential result.

Corollary 7.6.1. *Let $\mathcal{X}$ be a countable domain. Suppose that for some sample size n_0, there exists an (α_0, β_0)-accurate ρ_0-*TV *indistinguishable learner A for a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ with $\alpha_0 \in (0, 1/2), \rho_0 \in (0,1), \beta_0 \in \left(0, \frac{1-\rho_0}{1+\rho_0}\right)$. Then, for any $(\alpha, \beta, \rho) \in (0,1)^3$, $\mathcal{H}$ admits an (α, β)-accurate ρ-*TV *indistinguishable learner A'.*

This result relies on connections between learnability by TV indistinguishable learners and finiteness of the Littlestone dimension of the underlying hypothesis class that were discussed in Section 7.5. In particular, Corollary 7.10.17 shows that the existence of such a non-trivial TV indistinguishable learner implies that the $\mathcal{H}$ has finite Littlestone dimension, and Proposition 7.10.12, states that the finiteness of the Littlestone dimension of $\mathcal{H}$ implies the existence of an (α, β)-accurate ρ-TV indistinguishable learner, for arbitrarily small choices of α, β, ρ. It is not hard to see that we need to constrain $\alpha \in (0, 1/2)$, because the algorithm needs to have an advantage compared to the random classifier. Moreover, it should be the case that $\beta \in (0, 1-\rho)$. If $\beta \geq 1-\rho$ then the algorithm which outputs a constant classifier with probability β and an α-good one with the remaining probability is ρ-TV indistinguishable and (α, β)-accurate. An interesting open problem is to investigate what happens when $\beta \in \left(\frac{1-\rho}{1+\rho}, 1-\rho\right)$.

We underline that Corollary 7.6.1 is existential and does not make actual use of the weak TV indistinguishable learner that is given as input. Hence, it is natural to try to come up with sample-efficient and constructive approaches that utilize the weak learner through black-box oracle calls to it during the derivation of the strong one. In what follows, we aim to design such algorithms. We remind the reader that if we constrain ourselves to work in the setting where $\mathcal{X}$ is countable, then the absolute continuity requirement in the next theorems comes immediately, due to Claim 19.

Indistinguishability Amplification. We first consider the amplification of the indistinguishability guarantees of an algorithm. An important ingredient of our approach is a replicable algorithm for finding heavy hitters of a distribution, i.e., elements whose frequency is above some given threshold. This algorithm has appeared in (GKM21; ILPS22). However, the dependence of the number of samples in the confidence parameter in these works is polynomial. We present a new variant of this algorithm that has polylogarithmic dependence on the confidence parameter. Moreover, using a stronger concentration inequality, we improve the dependence of the number of samples on the error parameter. We believe that this result could be of independent interest. We also design an agnostic learner for finite hypothesis classes. However, the dependence of the number of samples on $|\mathcal{H}|$ is polynomial. We believe that an interesting question is to design agnostic learners with polylogarithmic dependence on $|\mathcal{H}|$. We refer the reader to Section 7.11.

Theorem 7.6.2 (Indistinguishability Amplification)**.** *Let $\mathcal{P}$ be a reference probability measure over $\{0,1\}^{\mathcal{X}}$ and $\mathcal{D}$ be a distribution over inputs. Consider the source of randomness $\mathcal{R}$ to be a Poisson point process with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, where* Leb *is the Lebesgue measure over $\mathbb{R}_+$. Consider a weak learning rule A that is (i) ρ-TV indistinguishable with respect to $\mathcal{D}$ for some $\rho \in (0,1)$, (ii) (α,β)-accurate for $\mathcal{D}$ for some $(\alpha,\beta) \in (0,1)^2$, such that $\beta < \frac{2\rho}{\rho+1} - 2\sqrt{\frac{2\rho}{\rho+1}} + 1$, and, (iii) absolutely continuous with respect to $\mathcal{P}$ on inputs from $\mathcal{D}$. Then, for any $\rho', \epsilon, \beta' \in (0,1)^3$, there exists a learner $\textsc{Ampl}(A, \mathcal{R}, \beta', \epsilon, \rho')$ that is ρ'-TV indistinguishable with respect to $\mathcal{D}$, and $(\alpha + \epsilon, \beta')$-accurate for $\mathcal{D}$.*

We remark that the above result makes strong use of the equivalence between replicability and TV indistinguishability. Our algorithm is a variant of the amplification algorithm that appeared in (ILPS22), which (i) works for a wider range of parameters and (ii) its sample complexity is polylogarithmic in the parameter β'.

Accuracy Boosting. Next, we design an algorithm that boosts the accuracy of an n-sample ρ-TV indistinguishable algorithm and preserves its TV indistinguishability guarantee. Our algorithm is a variant of the boosting mechanism provided in (ILPS22). Similarly as in the case of amplification, our variant improves upon the dependence of the number of samples on the parameter β'.

Theorem 7.6.3 (Accuracy Boosting)**.** *Let $\mathcal{P}$ be a reference probability measure over $\{0,1\}^{\mathcal{X}}$ and $\mathcal{D}$ be a distribution over inputs. Consider the source of randomness $\mathcal{R}$ to be a Poisson point process with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, where* Leb *is the Lebesgue measure over $\mathbb{R}_+$. Consider a weak learning rule A that is (i) ρ-TV* *indistinguishable with respect to $\mathcal{D}$ for some $\rho \in (0,1)$, (ii) $(1/2-\gamma,\beta)$-accurate for $\mathcal{D}$ for some $(\gamma,\beta) \in (0,1)^2$, and, (iii) absolutely continuous with respect to $\mathcal{P}$ on inputs from $\mathcal{D}$. Then, for any $\beta', \epsilon, \rho' \in (0,1)^3$, there exists a learner* $\textsc{Boost}(A, \mathcal{R}, \epsilon)$ *that is ρ'-TV indistinguishable with respect to $\mathcal{D}$ and (ϵ, β')-accurate for $\mathcal{D}$.*

We can combine the amplification and boosting results for a wide range of parameters and get the next corollary.

Corollary 7.6.4. *Let $\mathcal{X}$ be a countable domain and A be an n-sample ρ-TV indistinguishable (α,β)-accurate algorithm, for some $\rho \in (0,1), \alpha \in (0,1/2), \beta \in \left(0, \frac{2\rho}{\rho+1} - 2\sqrt{\frac{2\rho}{\rho+1}} + 1\right)$. Then, for any $\rho', \alpha', \beta' \in (0,1)^3$, there exists a ρ'-TV indistinguishable (α', β')-accurate learner A' that requires at most $O\left(\mathrm{poly}\left(1/\rho, 1/\alpha', \log(1/\beta')\right) \cdot n\right)$ samples from $\mathcal{D}$.*

The proof of this result follows immediately from Theorem 7.6.2, Theorem 7.6.3, and from the fact that we can design the reference probability measure $\mathcal{P}$ for countable domains (cf. Claim 19). This result leads to two natural questions: what is the tightest range of β for which we can amplify the stability parameter ρ and under what assumptions can we design such boosting and amplification algorithms for general domains $\mathcal{X}$? For a more detailed discussion, we refer the reader to Section 7.11.3, Section 7.11.4.

Remark 12 (Dependence on the Parameters)**.** *We underline that the polynomial dependence on ρ in the boosting result is not an artifact of the algorithmic procedure or the analysis we provide, but it is rather an inherent obstacle in TV indistinguishability. (ILPS22) show that in order to estimate the bias of a coin ρ-replicably with accuracy τ one needs at least $1/(\tau^2\rho^2)$ coin tosses. Since ρ-TV indistinguishability implies $(2\rho/(1+\rho))$-replicability as we have shown (without any blow-up in the sample complexity), we also inherit this lower bound. Our main goal behind the study of the boosting algorithms is to identify the widest range of parameters α, ρ, β such that coming up with a ρ-TV indistinguishable algorithm switches from being trivial to being difficult. For example, in PAC learning we know that if the accuracy parameter is strictly less than $1/2$, then there are sample-efficient boosting algorithms that can drive it down to any $\epsilon > 0$. In the setting we are studying, it is crucial to understand the relationship between β, ρ, see Section 7.11.3.*

7.7 Preliminaries and Additional Definitions

7.7.1 Preliminaries

Probability Theory. We first review some standard definitions from probability theory.

Definition 7.7.1 (Coupling)**.** *A coupling of two probability distributions P and Q is a pair of random variables (X, Y), defined on the same probability space, such that the marginal distribution of X is P and the marginal distribution of Y is Q.*

Definition 7.7.2 (Integral Probability Metric)**.** *The Integral Probability Metric (IPM) between two probability measures P and Q over $\mathcal{O}$ is defined as*

$$d_{\mathcal{F},\mathcal{O}}(P, Q) = \sup_{f \in \mathcal{F}} \left| \int_{\mathcal{O}} f dP - \int_{\mathcal{O}} f dQ \right| = \sup_{f \in \mathcal{F}} \left| \mathop{\mathbb{E}}_{x \sim P}[f(x)] - \mathop{\mathbb{E}}_{x \sim Q}[f(x)] \right| ,$$

where $\mathcal{F}$ is a set of real-valued bounded functions $\mathcal{O} \to \mathbb{R}$.

IPM distance measures are symmetric and non-negative. Note that the KL-divergence is not a special case of IPM, rather it belongs to the family of f-divergences, that intersect with IPM only at the TV distance. Such measures were recently used in order to derive PAC-Bayes style generalization bounds (AEMM22). The definition of an f-divergence will not be useful in this chapter and we refer the interested reader to e.g., (SV16).

Learning Theory. We next review some standard definitions in statistical learning theory. We start with the definition of the Littlestone dimension (Lit88).

Definition 7.7.3 (Littlestone Dimension (Lit88))**.** *Consider a complete binary tree T of depth $d+1$ whose internal nodes are labeled by points in $\mathcal{X}$ and edges by $\{0, 1\}$, when they connect the parent to the right, left child, respectively. We say that $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ Littlestone-shatters T if for every root-to-leaf path $x_1, y_1, \ldots, x_d, y_d, x_{d+1}$ there exists some $h \in \mathcal{H}$ such that $h(x_i) = y_i, 1 \leq i \leq d$. The Littlestone dimension is denoted by* $\mathrm{Ldim}(\mathcal{H})$ *is defined to be the largest d such that $\mathcal{H}$ Littlestone-shatters such a binary tree of depth $d + 1$. If this happens for every $d \in \mathbb{N}$ we say that* $\mathrm{Ldim}(\mathcal{H}) = \infty$.

We work under the well-known PAC learning model that was introduced in (Val84). Let us denote the misclassification probability of a classifier h by $\mathrm{err}_{\mathcal{D}}(h) = \mathbf{Pr}_{(x,y) \sim \mathcal{D}}[h(x) \neq y]$. Also, we say that $\mathcal{D}$ is realizable with respect to $\mathcal{H}$ if there exists some $h^* \in \mathcal{H}$ such that $\mathrm{err}_{\mathcal{D}}(h^*) = 0$. Below, we slightly abuse notation and use the misclassification probability for distributions over classifiers.

Definition 7.7.4 (PAC Learnability (Val84; SSBD14))**.** *An algorithm A is n-sample (α, β)-accurate for a hypothesis class $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ if, for any realizable distribution $\mathcal{D}$, it holds that $\mathbf{Pr}_{S \sim \mathcal{D}^n}[\mathrm{err}_{\mathcal{D}}(A(S)) > \alpha] \leq \beta$. A hypothesis class $\mathcal{H}$ is PAC learnable if, for any $\alpha, \beta \in (0, 1)^2$, there exist some $n_0(\alpha, \beta) \in \mathbb{N}$ and an algorithm A such that A is n-sample (α, β)-accurate for $\mathcal{H}$, for any $n \geq n_0(\alpha, \beta)$.*

For the purposes of this work, an algorithm A should be thought of as a mapping from samples to a *distribution* over hypotheses. We want to design algorithms that satisfy two desiderata: they are PAC learners for some given hypothesis class $\mathcal{H}$ and they are total variation indistinguishable. In particular, we consider the following learning setting combining Definition 7.1.3 and 7.7.4.

Definition 7.7.5 (Realizable Learnability by TV Indistinguishable Learner)**.** *An algorithm A is n-sample (α, β)-accurate ρ-*TV *indistinguishable for a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ if, for any realizable distribution $\mathcal{D}$, it holds that (i) A is n-sample ρ-*TV *indistinguishable and (ii) $\mathbf{Pr}_{S\sim\mathcal{D}^n}[\mathrm{err}_{\mathcal{D}}(A(S)) > \alpha] \leq \beta$. A hypothesis class $\mathcal{H}$ is learnable by a* TV *indistinguishable algorithm if, for any $\alpha, \beta, \rho \in (0,1)$, there exist some $n_0(\alpha, \beta, \rho) \in \mathbb{N}$ and an algorithm A such that A is n-sample (α, β)-accurate ρ-*TV *indistinguishable for $\mathcal{H}$ for any $n \geq n_0(\alpha, \beta, \rho)$.*

In the above definition, n depends on α, β, ρ (and $\mathcal{H}$), but *not* on the distribution.

Definition 7.7.6 (Uniform Convergence Property)**.** *We say that a domain $\mathcal{X}$ and a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ satisfy the uniform convergence property if there exists a function $m^{\mathrm{UC}} : (0,1)^2 \to \mathbb{N}$ such that for any $\epsilon, \delta \in (0,1)$, and for every distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ it holds that if $S \sim \mathcal{D}^m$ and $m \geq m^{\mathrm{UC}}(\epsilon, \delta)$, it holds that $\sup_{h\in\mathcal{H}} |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon$, with probability at least $1-\delta$, where L_S (resp. $L_{\mathcal{D}}$) is the empirical (resp. population) loss.*

The fundamental theorem of learning theory (VC15; BEHW89) states that the uniform convergence property is equivalent to the finiteness of the VC dimension of $\mathcal{H}$. However, one needs to make some (standard) measurability assumptions on $\mathcal{X}, \mathcal{H}$ to rule out pathological cases. For instance, it is known that there classes with VC dimension 1 where uniform convergence does not hold (BD15)[5]. It is known that when $\mathcal{H}$ is countable and has finite VC dimension uniform convergence holds (BM02).

7.7.2 General Definition of Indistinguishability

While in the main body of the paper, we focused on binary classification, (most of) our proofs extend to general learning problems and so we first present a general abstract framework.

For general learning tasks, we can view learning rules (or algorithms) as randomized mappings $A : \mathcal{I} \to \Delta_{\mathcal{O}}$ which take as input instances from a domain $\mathcal{I}$ and map them to an element of the output space $\mathcal{O}$. We assume that there is a distribution μ on $\mathcal{I}$ that generates instances.

A second way to view the learning algorithm is via the mapping $A : \mathcal{I} \times \mathcal{R} \to \mathcal{O}$. Then A takes as input an instance $I \sim \mu$ and a random string $r \sim \mathcal{R}$ (we use $\mathcal{R}$ for both the probability space and the distribution) corresponding to the algorithm's *internal randomness* and outputs $A(I, r) \in \mathcal{O}$. Thus, $A(I)$ is a distribution over $\mathcal{O}$ whose randomness comes from the random variable r, while $A(I, r)$ is a deterministic object.

The space $\Delta_{\mathcal{O}}$ is endowed with some statistical dissimilarity measure.

[5] We note that the proof of the existence of such a class holds under the continuum hypothesis.

Definition 7.7.7 (Indistinguishability)**.** *Let $\mathcal{I}$ be an input space, $\mathcal{O}$ be an output space and d be some statistical dissimilarity measure. A learning rule A satisfies ρ-indistinguishability with respect to d if for any distribution μ over $\mathcal{I}$ and two independent instances $I, I' \sim \mu$, it holds that*

$$\mathop{\mathbb{E}}_{I,I'\sim\mu}\left[d\left(A(I), A(I')\right)\right] \leq \rho\,.$$

To illustrate the generality of our definition, we now show how we can instantiate $\mathcal{I}, \mathcal{O}, \mu, d$ to recover other definitions about stability of learning algorithms appearing in prior work.

Global Stability. Global stability (BLM20) is a fundamental property of learning algorithms that was recently used to establish an equivalence between online learnability and approximate differential privacy in binary classification. We show how we can recover the definition of global stability. Let us first recall the definition.

Definition 7.7.8 (Global Stability (BLM20))**.** *Let $\mathcal{R}$ be a distribution over random strings. A learning rule A is n-sample η-globally stable if for any distribution $\mathcal{D}$ there exists a hypothesis $h_{\mathcal{D}}$ such that*

$$\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n, r\sim\mathcal{R}}[A(S,r) = h_{\mathcal{D}}] \geq \eta\,.$$

In order to recover Definition 7.7.8 using Definition 7.7.7 we let $(S, r) \in \mathcal{I}, \mu = \mathcal{D}^n \times \mathcal{R}$ and $d(A(I,r), A(I',r')) = \mathbb{1}_{A(I,r)\neq A(I',r')}$. Thus, we have that

$$\mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n, r,r'\sim\mathcal{R}}[\mathbb{1}_{A(S,r)\neq A(S',r')}] \leq \rho \implies$$
$$\mathop{\mathbf{Pr}}_{S,S'\sim\mathcal{D}^n, r,r'\sim\mathcal{R}}[A(S,r) \neq A(S',r')] \leq \rho.$$

Notice that this gives us a two-sided version of the definition of global-stability. So far we have established that $\mathbf{Pr}_{S,S'\sim\mu,r,r'\sim\mathcal{R}}[A(S,r) = A(S',r')] \geq 1-\rho > 0$. Since two independent draws of the random variable $A(S,r)$ are the same with non-zero probability it means that it must have point masses. Moreover, there are countably many such point masses. Let $\mathcal{H}_m = \{h \in \mathcal{H} : \mathbf{Pr}_{S\sim\mathcal{D}^n,r\sim\mathcal{R}}[A(S,r) = h]\}$. Then,

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{S,S'\sim\mu,r,r'\sim\mathcal{R}}[A(S,r) = A(S',r')] &= \sum_{h\in\mathcal{H}_m}\left(\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n,r\sim\mathcal{R}}[A(S,r) = h]\right)^2 \\
&\leq \max_{h\in\mathcal{H}_m}\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n,r\sim\mathcal{R}}[A(S,r) = h]\cdot\sum_{h\in\mathcal{H}_m}\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n,r\sim\mathcal{R}}[A(S,r) = h] \\
&\leq \max_{h\in\mathcal{H}_m}\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n,r\sim\mathcal{R}}[A(S,r) = h] \\
&= \max_{h\in\mathcal{H}}\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n,r\sim\mathcal{R}}[A(S,r) = h]
\end{aligned}$$

Thus, by chaining the two inequalities we have established, we get that $\max_{h\in\mathcal{H}_m}\mathbf{Pr}_{S\sim\mathcal{D}^n,r\sim\mathcal{R}}[A(S,r) = h] \geq 1-\rho$, so the algorithm A satisfies the notion of global stability.

7.7.3 Alternative Definitions of TV Indistinguishability

We now discuss alternative ways to define TV indistinguishability.

TV Indistinguishability with Fixed Prior

First, observe that the definition we propose is two-sided in the sense that we require drawing two sets of i.i.d. samples. A different way to view TV indistinguishability is by requiring that the output of the algorithm is close, in TV distance, to some *prior distribution*, which depends on the data-generating process $\mathcal{D}$ but is independent of the sample. Notice that we could introduce a similar one-sided general definition as a second viewpoint of Definition 2.4.6 (named Indistinguishability with Fixed Prior).

Definition 7.7.9 (TV Indistinguishability with Fixed Prior)**.** *A learning rule A is n-sample ρ-fixed prior* TV *indistinguishable if for any distribution over inputs $\mathcal{D}$, there exists some prior $\mathcal{P}_{\mathcal{D}}$ such that for $S \sim \mathcal{D}^n$ it holds that*

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}[d_{\mathrm{TV}}(A(S), \mathcal{P}_{\mathcal{D}})] \leq \rho\,.$$

Notice that, using the triangle inequality, we can see that this definition is equivalent to Definition 7.1.3, up to a factor of 2. Formally, we have the following result.

Lemma 7.7.10. *If A is ρ-*TV *indistinguishable then it is ρ-fixed prior* TV *indistinguishable. Conversely, if A is ρ-fixed prior* TV *indistinguishable then it is* 2ρ-TV *indistinguishable.*

We remark that if A is TV indistinguishable with respect to a distribution over inputs $\mathcal{D}$, one can show that it is also fixed prior TV indistinguishable with respect to $\mathcal{D}$ where the fixed prior is equal to $\mathcal{P}_{\mathcal{D}} = \int_S A(S) d(\mathcal{D}^n)$.

Proof. For the first direction, we let $\mathcal{P}_{S,S'}$ be a distribution with the property that $d_{\mathrm{TV}}(A(S), \mathcal{P}_{S,S'}) = d_{\mathrm{TV}}(A(S'), \mathcal{P}_{S,S'}) = d_{\mathrm{TV}}(A(S), A(S'))/2$, e.g., $\mathcal{P}_{S,S'} = 1/2 \cdot (A(S) + A(S'))$, for every $S, S' \sim \mathcal{D}^n$. We now define $\mathcal{P}_{\mathcal{D}}$ to be the average of $\mathcal{P}_{S,S'}$ with respect to the measure of the product distribution of S, S'. We have that

$$\begin{aligned}
\mathcal{P}_{\mathcal{D}} &= \int_{S,S'} \mathcal{D}^n(S)\mathcal{D}^n(S') \frac{A(S) + A(S')}{2} dS dS' \\
&= \int_T \left(\mathcal{D}^n(T) 1\{S = T\} \frac{A(T)}{2} \left(\int_{S'} \mathcal{D}^n(S') \right) + \mathcal{D}^n(T) 1\{S' = T\} \frac{A(T)}{2} \left(\int_S \mathcal{D}^n(S) \right) \right) dS dS' = \\
&= \int_T \mathcal{D}^n(T) A(T) dT\,.
\end{aligned}$$

This means that $\mathbb{E}_{S \sim \mathcal{D}^n}[d_{\mathrm{TV}}(A(S), \mathcal{P}_{\mathcal{D}})] = \int_S \mathcal{D}^n(S) d_{\mathrm{TV}} \left(A(S), \int_T \mathcal{D}^n(T) A(T) dT \right) dS \leq \rho$.

For the converse, notice that

$$\begin{aligned}\mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S),A(S'))] &\leq \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S),\mathcal{P}_{\mathcal{D}})+d_{\mathrm{TV}}(A(S'),\mathcal{P}_{\mathcal{D}})] \\ &= \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S),\mathcal{P}_{\mathcal{D}})] + \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S'),\mathcal{P}_{\mathcal{D}})] \\ &= 2\mathop{\mathbb{E}}_{S\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S),\mathcal{P}_{\mathcal{D}})] \\ &\leq 2\rho.\end{aligned}$$

□

With High Probability TV Indistinguishability

A different direction in which we can extend the definition of total variation indistinguishability has to do with replacing the expectation with a high-probability style of bound. We remark that (ILPS22) provide a similar alternative definition in the context of their work.

Definition 7.7.11 (High-Probability TV Indistinguishability)**.** *A learning rule A is n-sample high-probability (η,ν)-*TV *indistinguishable if for any distribution $\mathcal{D}$ there exists some prior $\mathcal{P}_{\mathcal{D}}$ such that*

$$\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S),\mathcal{P}_{\mathcal{D}})\leq\eta]\geq 1-\nu\,.$$

Notice that in the above definition we have used the fixed prior version of TV indistinguishability to reduce the number of parameters, but it can also be stated in its the two-sided version. It is not hard to see that the "in expectation" and the "with high probability" versions of the definition are qualitatively equivalent. Moreover, we can establish a quantitative connection as follows.

Lemma 7.7.12. *If a learning rule A is an n-sample ρ-fixed prior* TV *indistinguishable learner (cf. Definition 7.7.9) then it is an n-sample high-probability $(\rho/\nu,\nu)$-*TV *indistinguishable learning rule (cf. Definition 7.7.11), for any $\rho\leq\nu<1$. Conversely, if a learnigng rule A is an n-sample high-probability (η,ν)-*TV *indistinguishable learner then it is an n-sample $(\eta+\nu-\eta\cdot\nu)$-fixed prior* TV *indistinguishable learning rule.*

Proof. The proof of the first part of claim is a direct consequence of Markov's inequality. Notice that $d_{\mathrm{TV}}(A(S),\mathcal{P}_{\mathcal{D}})$ is random variable whose expected value is bounded by ρ. Thus, we have that

$$\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S),\mathcal{P}_{\mathcal{D}})\geq\rho/\nu]\leq\nu\,.$$

Hence, we can see that A is a high-probability $(\rho/\nu,\nu)$-TV indistinguishable learning rule.

We now move to the second part of the claim. Let $\mathcal{E}$ be the event that $d_{\mathrm{TV}}(A(S), \mathcal{P}_{\mathcal{D}}) \geq \eta$. Then, we have that

$$\begin{aligned}\mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}[d_{\mathrm{TV}}(A(S), \mathcal{P}_{\mathcal{D}})] &= \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}[d_{\mathrm{TV}}(A(S), \mathcal{P}_{\mathcal{D}}) | \mathcal{E}] \, \mathbf{Pr}[\mathcal{E}] + \mathop{\mathbb{E}}_{S \sim \mathcal{D}^n}[d_{\mathrm{TV}}(A(S), \mathcal{P}_{\mathcal{D}}) | \mathcal{E}^c] \, \mathbf{Pr}[\mathcal{E}^c] \\ &\leq 1 \cdot \nu + \eta \cdot (1 - \nu) \\ &= \eta + \nu - \eta \cdot \nu. \end{aligned}$$

□

7.7.4 Coupling and Correlated Sampling

Coupling is a fundamental notion in probability theory with many applications (LP17b). The correlated sampling problem, which has applications in various domains, e.g., in sketching and approximation algorithms (Bro97; Cha02), is described in (BGH$^+$16) as follows: Alice and Bob are given probability distributions P and Q, respectively, over a finite set Ω. *Without any communication, using only shared randomness* as the means to coordinate, Alice is required to output an element x distributed according to P and Bob is required to output an element y distributed according to Q. Their goal is to minimize the disagreement probability $\mathbf{Pr}[x \neq y]$, which is comparable with $d_{\mathrm{TV}}(P, Q)$. Formally, a correlated sampling strategy for a finite set Ω with error $\epsilon : [0,1] \to [0,1]$ is specified by a probability space $\mathcal{R}$ and a pair of functions $f, g : \Delta_\Omega \times \mathcal{R} \to \Omega$, which are measurable in their second argument, such that for any pair $P, Q \in \Delta_\Omega$ with $d_{\mathrm{TV}}(P, Q) \leq \delta$, it holds that (i) the push-forward measure $\{f(P, r)\}_{r \sim \mathcal{R}}$ (resp. $\{g(Q, r)\}_{r \sim \mathcal{R}}$) is P (resp. Q) and (ii) $\mathbf{Pr}_{r \sim \mathcal{R}}[f(P, r) \neq g(Q, r)] \leq \epsilon(\delta)$. We underline that a correlated sampling strategy is *not* the same as a coupling, in the sense that the latter requires a single function $h : \Delta_\Omega \times \Delta_\Omega \to \Delta_{\Omega \times \Omega}$ such that for any P, Q, the marginals of $h(P, Q)$ are P and Q respectively. It is known that for any coupling function h, it holds that $\mathbf{Pr}_{(x,y) \sim h(P,Q)}[x \neq y] \geq d_{\mathrm{TV}}(P, Q)$ and that this bound is attainable. Since $\{(f(P, r), g(Q, r))\}_{r \sim \mathcal{R}}$ induces a coupling, it holds that $\epsilon(\delta) \geq \delta$ and, perhaps surprisingly, there exists a strategy with $\epsilon(\delta) \leq \frac{2\delta}{1+\delta}$ (Bro97; KT02; Hol07) and this result is tight (BGH$^+$16). A second difference between coupling and correlated sampling has to do with the size of Ω: while correlated sampling strategies can be extended to infinite spaces Ω, it remains open whether there exists a correlated sampling strategy for general measure spaces $(\Omega, \mathcal{F}, \mu)$ with any non-trivial error bound (BGH$^+$16). On the other hand, coupling applies to spaces Ω of any size.

(GKM21) studied user-level privacy and introduced the notion of pseudo-global stability, which is essentially the same as replicability as observed by (ILPS22). (GKM21) showed that pseudo-global stability is qualitatively equivalent to approximate differential privacy. Their main technique was the use of correlated sampling that allowed users to output the same learned hypothesis (stability) employing shared randomness. We mention that (GKM21) provide their results for finite outcome space (i.e., $\mathcal{X}$ is finite and thus $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is too). In particular, they need finiteness of the domain in order to apply correlated sampling which is used during their "DP implies pseudo-global stability" reduction. They mention that their results can be extended to the case where $\mathcal{X}$ is infinite and that this does require non-trivial generalization of tools such as correlated sampling and some measure-theoretic details to that setting[6]; we refer to a discussion in Section 5.3 of (BGH+16) about the assumptions needed in order to achieve correlated sampling in infinite spaces. Similarly, the last step of the constructive transformation of a DP algorithm to a replicable one provided in (BGH+23) uses correlated sampling and is hence also given for finite domains. For further comparisons between our coupling and the correlated sampling problem of (BGH+16), we refer to the discussion in (AS19) after Corollary 4.

A very useful tool for our derivations is a coupling protocol that can be found in (AS19).

Theorem 7.7.13 (Pairwise Optimal Coupling (AS19))**.** *Let $\mathcal{S}$ be any collection of random variables that are absolutely continuous with respect to a common probability measure[7] μ. Then, there exists a coupling of the variables in $\mathcal{S}$ such that, for any $X, Y \in \mathcal{S}$,*

$$\mathbf{Pr}[X \neq Y] \leq \frac{2d_{\mathrm{TV}}(X,Y)}{1 + d_{\mathrm{TV}}(X,Y)}\,.$$

Moreover, this coupling requires sample access to a Poisson point process with intensity $\mu \times \mathrm{Leb} \times \mathrm{Leb}$, where Leb is the Lebesgue measure over $\mathbb{R}_+$, and full access to the densities of all the random variables in $\mathcal{S}$ with respect to μ.

An intuitive illustration of how it works can be found in Figure 7.1.

7.7.5 Discussion on Definition 7.1.3

We discuss more extensively the TV Indistinguishability definition. One important motivation for the definition of TV indistinguishability is to show that replicability can be equivalently defined using the same high-level template like the well-studied PAC-Bayes framework, where one shows that the outputs of the algorithms are close, under the KL divergence, with some data-independent priors. In other words, our results show how to organize and view different well-studied notions of stability using the same template.

Moreover, an interpretation of the replicability definition is that two executions of the algorithm over independent datasets should be coupled using just shared internal randomness. However, this is one of potentially infinite ways to couple

[6] To be more specific, the proof of Theorem 20 in (GKM21) requires to define the correlated sampling strategy over the space $2^{\mathcal{X}}$ a priori (independently of the observed samples and input algorithm). Hence while the strategy is applied to distributions with finite support, an extension to infinite domain in that proof would require some modifications.

[7] This result extends to the setting where μ is a σ-finite measure, but it is not needed for the purposes of our work.

the two executions. Our definition, which we find quite natural, captures exactly this observation and allows for general couplings between two random runs. It is also worth noting that, to the best of our knowledge, all the notions of algorithmic stability that have been proposed in the past do not depend on the source of internal randomness of the algorithm. However, this is not the case with replicability.

Let us now present a concrete algorithm whose stability property is easier to prove under the new definition. (GKM21) presented a procedure that transforms a list-globally stable algorithm to a replicable one (Algorithm 1, page 9 in (GKM21)). Crucially, in the last step of this algorithm the authors use a correlated sampling procedure to prove the replicability property. This procedure induces a computational overhead to the overall algorithm, and it is not clear even if it is computable beyond finite domains. On the other hand, the TV indistinguishability property is immediate. Thus, the transformation from list-global stability to TV indistinguishability is computationally efficient and holds for general domains whereas the transformation from list-global stability to replicability is not.

To the best of our knowledge, most of the replicable algorithms that have been developed use their internal randomness over data-independent distributions. To make this point more clear let us consider the replicable SQ oracle of (ILPS22). In this work, the authors use randomness over distributions that are independent of the input sample S. Thus, no matter how the internal randomness is implemented, when one shares it across two executions the internal random choices of the algorithm are the same.

However, there are algorithms, like Algorithm 1 in (GKM21), that use internal randomness over a data-dependent distribution. If the algorithm makes random choices over data-dependent quantities like in (GKM21), when one shares the randomness across two executions the internal random choices are not necessarily the same even if the TV distance between the two distributions is small, unless one specifies carefully the source of internal randomness (i.e., using some coupling). This can lead to significant computational overhead when the domain is finite, computability issues when the domain is countable, and for general domains it is not clear yet that going from TV indistinguishability to replicability is possible. Hence, one advantage of TV indistinguishability is that it provides a relaxation over the stronger definition of replicability, which is the notion that our definition builds upon.

7.8 Useful Replicable Subroutines

In this section we present various replicable subroutines that will be useful in the derivation of our results.

7.8.1 Replicability Preliminaries

Recall the Statistical Query (SQ) model that was introduced by (Kea98) and is a restriction of the PAC learning model, appearing in various learning theory contexts (BKW03; GHRU11; CKMY20; GGK20; FKKT21). In the SQ model, the learner interacts with an oracle in the following way: the learner submits a statistical query to the oracle and the oracle returns its expected value, after adding some noise to it. More formally, we have the following definition.

Definition 7.8.1 ((Kea98)). *Let $\tau, \delta \in (0,1)^2, \mathcal{D}$ be a distribution over the domain $\mathcal{X}$ and $\phi : \mathcal{X} \to [0,1]$ be a query. Let S be an i.i.d. sample of size $n = n(\tau, \delta)$. Then, the statistical query oracle outputs a value v such that $|v - \mathbb{E}_{x\sim\mathcal{D}}[\phi(x)]| \leq \tau$, with probability at least $1 - \delta$.*

Essentially, using a large enough number of samples, the SQ oracle returns an approximation of the expected value of a statistical query whose range is bounded. (ILPS22) provide a replicable implementation of an SQ oracle with a mild blow-up in the sample complexity.

Theorem 7.8.2 (Replicable SQ Learner (ILPS22)). *Let $\tau, \delta, \rho \in (0,1)^3, \delta \leq \rho/3, \mathcal{D}$ be a distribution over some domain $\mathcal{X}$, and $\phi : \mathcal{X} \to [0,1]$ be a query. Let S be an i.i.d. sample of size*

$$n = O\left(\frac{1}{\tau^2\rho^2}\log(1/\delta)\right).$$

Then there exists a ρ-replicable SQ oracle for ϕ.

The interpretation of the previous theorem is that we can estimate replicably statistical queries whose range is bounded.

The following result that was proved in (ILPS22) is useful for our derivations.

Claim 20 (ρ-Replicability $\implies$ (η, ν)-Replicability (ILPS22)). *Let A be a ρ-replicable algorithm and $\mathcal{R}$ be its source of randomness. Then for any $\nu \in [\rho, 1)$, it holds that*

$$\mathop{\mathbf{Pr}}_{r\sim\mathcal{R}}\left[\left\{\exists h \in \mathcal{H} : \mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n}[A(S,r) = h] \geq 1 - \frac{\rho}{\nu}\right\}\right] \geq 1 - \nu\,.$$

Notice that in the definition of replicability (Definition 7.1.1), the learner shares all the internal random bits across its two executions. A natural extension is to consider learners that share only *part* of their random bits, i.e., they have access to private random bits that are not shared across its executions and public random bits that are shared. A result in (ILPS22) shows that these learners are, essentially, equivalent to the ones that use only private bits. To be more precise, we say that a learner A is ρ-replicable with respect to r_{pub} if

$$\mathop{\mathbf{Pr}}_{S,S'\sim\mathcal{D}^n, r_{priv}, r'_{priv}, r_{pub}\sim\mathcal{R}}[A(S, r_{priv}, r_{pub}) = A(S', r'_{priv}, r_{pub})] \geq 1 - \rho\,.$$

The following result states this property formally.

Lemma 7.8.3 (Public, Private Replicability $\implies$ Replicability (ILPS22))**.** *Let A be an n-sample ρ-replicable learner with respect to r_{pub}. Then, A is a n-sample ρ-replicable learner with respect to (r_{pub}, r_{priv}).*

This result allows us to think of a replicable learner as having access to two different sources of randomness, one that is private to its execution and one that is shared across the executions. We will make use of it in transformations from DP learners to replicable learners and some boosting results.

7.8.2 Replicable Heavy-Hitters

In the analysis of the replicable heavy-hitter algorithm (cf. Algorithm 12) we will use the Bretagnolle-Huber-Carol inequality that bounds the estimation error of the parameters of a multinomial distribution from samples.

Lemma 7.8.4 (Bretagnolle-Huber-Carol Inequality (VW97))**.** *Let $p = (p_1, \ldots, p_k)$ multinomial distribution supported on k elements. Then, given access to n i.i.d. samples from p we have that*

$$\mathbf{Pr}\left[\sum_{i=1}^{k} |\widehat{p}_i - p_i| \geq \varepsilon\right] \leq 2^k e^{-n\varepsilon^2/2},$$

for every $\varepsilon \in (0,1)$, where $\widehat{p}_i$ is the empirical frequency of item i in the sample S.

The replicable heavy-hitters algorithm is depicted in Algorithm 12. As we alluded before, this approach is very similar to (GKM21; ILPS22). However, in our approach we treat the confidence parameter and the reproducibility parameters differently. Moreover, since we make use of Lemma 7.8.4, we are able to reduce the sample complexity of the algorithm.

Lemma 7.8.5. *Let $\mathcal{D}$ be distribution supported on some domain $\mathcal{X}$ and denote by $\mathcal{D}(x)$ the mass that it puts on $x \in \mathcal{X}$. For any $\epsilon, \delta, \rho, v \in (0,1)^4$ such that $(v - \epsilon, v + \epsilon) \subseteq (0,1)$, Algorithm 12 is ρ-replicable and outputs a list L such that, with probability $1 - \delta$, for all $x \in \mathcal{X}$:*

- *If $\mathcal{D}(x) < v - \epsilon$ then $x \notin L$.*
- *If $\mathcal{D}(x) > v + \epsilon$ then $x \in L$.*

Its sample complexity is at most $O\left(\frac{\log(1/(\min\{\delta,\rho\}(v-\varepsilon)))}{(v-\varepsilon)\rho^2\varepsilon^2}\right)$.

Proof. We first prove the correctness of the algorithm with the desired accuracy ε and confidence δ. For simplicity, let us assume that $\delta \leq \rho/4$. Otherwise, we can simply set $\delta = \rho/4$. After we pick n_1 points, the probability that a $(v - \varepsilon)$-heavy-hitter of the distribution is not included in S_1 is at most

$$(1 - (v - \varepsilon))^{n_1} \leq e^{-(v-\varepsilon)\cdot n_1} \leq \frac{\delta \cdot (v - \varepsilon)}{2}.$$

Algorithm 12 Replicable Heavy-Hitters

```
1: Input: Sample access to a distribution $\mathcal{D}$ over some domain $\mathcal{X}$
2: Parameters: Threshold $v$, error $\epsilon$, confidence $\delta$, replicability $\rho$
3: Output: List of elements $L$ in $\mathcal{X}$
4: $n_1 \leftarrow \frac{\log(2/(\min\{\delta,\rho\}(v-\varepsilon)))}{v-\varepsilon}$
5: $S_1 \leftarrow n_1$ i.i.d. samples. from $\mathcal{D}$
6: $\mathcal{X}_h \leftarrow$ unique elements of $S_1$          ▷ Notice that $|\mathcal{X}_h| \leq n_1$.
7: $n_2 \leftarrow \frac{32(\log(2/\min\{\delta,\rho\})+|\mathcal{X}|+1)}{\rho^2\epsilon^2}$
8: $S_2 \leftarrow n_2$ i.i.d. samples from $\mathcal{D}$
9: $\widehat{p}_x \leftarrow \mathrm{freq}_S(x), \forall x \in \mathcal{X}_h$          ▷ $\widehat{p}_x$ is the empirical frequency of every potential heavy hitter
10: $v' \leftarrow U[v-\varepsilon/2, v+\varepsilon/2]$          ▷ Set the threshold for acceptance of a heavy-hitter.
11: $L \leftarrow \{x \in \mathcal{X}_h : \widehat{p}_x \geq v'\}$          ▷ Drop the elements of $\mathcal{X}_h$ that fall below the threshold.
12: Output $L$
```

Since there are at most $1/(v-\varepsilon)$ such heavy-hitters, we can see that with probability at least $\delta/2$ all of the are included in S_1. Let us call this event $\mathcal{E}_1$ and condition on it for the rest of the proof.

Let us consider a distribution $\widehat{\mathcal{D}}$ that puts the same mass on every element of $\mathcal{X}_h$ as $\mathcal{D}$ and the remaining mass on a new special element e. We can sample from $\widehat{\mathcal{D}}$ in the following way: we draw a sample from $\mathcal{D}$ and if it falls in $\mathcal{X}_h$ we return it, otherwise we return e. Thus, we can see that if we draw n samples from $\widehat{\mathcal{D}}$, they are distributed according to a multinomial distribution supported on $\mathcal{X}_h \cup \{e\}$. Thus, Lemma 7.8.4 applies to this setting which means that if we draw n_2 i.i.d. samples from $\widehat{\mathcal{D}}$ we have that

$$\mathbf{Pr}\left[\sum_{i=1}^{k} |\widehat{p}_i - p_i| \geq \frac{\varepsilon\rho}{4}\right] \leq 2^k e^{-n_2\varepsilon^2\rho^2/32} \leq e^k e^{-n_2\varepsilon^2\rho^2/32} = e^{k-n_2\varepsilon^2\rho^2/32},$$

where $k = |\mathcal{X}_h| + 1$. Thus, $e^{k-n_2\varepsilon^2\rho^2/32} = e^{-\log(2/\delta)} \leq \frac{\delta}{2}$. We call this event $\mathcal{E}_2$ and condition on it for the rest of the proof. Notice that under this event we have that $|\widehat{p}_x - p_x| \leq \frac{\epsilon\rho}{4} < \frac{\epsilon}{2}, \forall x \in \mathcal{X}_h$. Since $v' \geq v - \epsilon/2$ it means that if $\widehat{p}_x \geq v' \geq v - \epsilon/2 \implies p_x + \epsilon/2 > v - \epsilon/2 \implies p_x > v - \epsilon$. Similarly, we get that if $\widehat{p}_x < v' \implies p_x < v + \epsilon$. Hence, we see that the algorithm is correct with probability at least $1 - \delta/2 - \delta/2 = 1 - \delta$. This concludes the correctness proof.

We now focus on the replicability of the algorithm. Let $\mathcal{X}_h^1$ be the unique elements at Line 6 of the algorithm in the first run and $\mathcal{X}_h^2$ in the second run. Notice that if $x \in (\mathcal{X}_h^1 \setminus \mathcal{X}_h^2) \cup (\mathcal{X}_h^2 \setminus \mathcal{X}_h^1)$ then, with probability at least $1-\delta/2-\delta/2 = 1-\delta$, the element x is not a $(v-\epsilon)$-heavy-hitter, so, with probability at least $1-\delta/2$, it will not be included in the output of the execution that it appears in. Let $E = \mathcal{X}_h^1 \cap \mathcal{X}_h^2$ and denote by L_1, L_2, the outputs of the first, second execution, respectively. We need to bound the probability of the event $\mathcal{E} = \{\exists x \in E : x \in L_1 \setminus L_2 \cup L_2 \setminus L_1\}$. Let $\widehat{p}_x^1, \widehat{p}_x^2$ the empirical frequencies of x in the first, second execution, respectively. Due to the concentration inequality we have used, we have that

$$\sum_{x \in \mathcal{X}_1 \cap \mathcal{X}_2} |\widehat{p}_x^i - p_x| \leq \frac{\varepsilon \rho}{4}, i \in \{1, 2\},$$

with probability at least $1-\delta$. Under this event, using the triangle inequality, this means that

$$\sum_{x \in \mathcal{X}_1 \cap \mathcal{X}_2} |\widehat{p}_x^1 - \widehat{p}_x^2| \leq \frac{\varepsilon \rho}{2}, i \in \{1, 2\},$$

Notice that since pick a number uniformly at random from an interval with range ϵ, for some given $x \in \mathcal{X}_1 \cap \mathcal{X}_2$, we have that $\mathbf{Pr}[x \in L_1 \setminus L_2 \cup L_2 \setminus L_1] \leq |\widehat{p}_x^1 - \widehat{p}_x^2|/\epsilon$. Thus, taking a union bound over $x \in \mathcal{X}_1 \cap \mathcal{X}_2$, we see that

$$\mathbf{Pr}[\mathcal{E}] \leq \frac{\sum_{x \in \mathcal{X}_1 \cap \mathcal{X}_2} |\widehat{p}_x^1 - \widehat{p}_x^2|}{2\epsilon} \leq \frac{\epsilon \rho}{2\epsilon} = \frac{\rho}{2}.$$

Putting everything together, we see that the probability that the two outputs of the algorithm differ is at most $\delta + \delta/2 + \rho/2 < \rho$. □

7.8.3 Replicable Agnostic PAC Learner for Finite $\mathcal{H}$

In this section we present a replicable agnostic PAC learner for finite hypothesis classes, i.e., a learner whose output is a hypothesis that has error rate close to the best one in the class. Our construction relies on the replicable SQ oracle from (ILPS22) (see Theorem 7.8.2). The idea is simple: since the error rate of every $h \in \mathcal{H}$ can be replicably estimated using Theorem 7.8.2, we do that for every $h \in \mathcal{H}$ and then we return the one that has the smallest estimated value.

Algorithm 13 Replicable Agnostic Learner for Finite $\mathcal{H}$

```
1: Input: Hypothesis class $\mathcal{H}$, sample access to a distribution
   $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$
2: Parameters: accuracy $\epsilon$, confidence $\delta$, replicability $\rho$
3: Output: Classifier $h$ that is $\epsilon$-close to the best one in $\mathcal{H}$
   and its estimated error on $\mathcal{D}$
4: $\widehat{\alpha}_h \leftarrow$ ReprErrorEst$(\epsilon/2, \delta/|\mathcal{H}|, \rho/|\mathcal{H}|), \forall h \in \mathcal{H}$    ▷ Theorem 7.8.2.
5: $\widehat{h}^* \leftarrow \arg\min_{h \in \mathcal{H}} \widehat{a}_h$    ▷ Break ties arbitrarily in a consistent manner.
6: Output $(\widehat{h}^*, \widehat{\alpha}_{\widehat{h}^*})$
```

It is not hard to see that Algorithm 13 is ρ-replicable and returns a hypothesis whose error is ϵ-close to the best one.

Claim 21. *Let $\mathcal{H}$ be a finite hypothesis class and $\epsilon, \delta, \rho \in (0,1)^3$. Given $O\left(\frac{|\mathcal{H}|^3}{\epsilon^2\rho^2}\log\left(\frac{|\mathcal{H}|}{\delta}\right)\right)$ i.i.d. samples from $\mathcal{D}$, Algorithm 13 is ρ-replicable and returns a classifier $\widehat{h}^*$ with* $\mathrm{err}(\widehat{h}^*) < \min_{h\in\mathcal{H}} \mathrm{err}(h) + \epsilon$, *with probability at least $1-\delta$.*

Proof. The replicability of the algorithm follows from the fact that we estimate each $\widehat{a}_h$ replicably with parameter $\rho/|\mathcal{H}|$ and we make $|\mathcal{H}|$ such calls.

Notice that for each call to the replicable error estimator we need $n_h = O\left(\frac{|\mathcal{H}|^2}{\epsilon^2\rho^2}\log\left(\frac{|\mathcal{H}|}{\delta}\right)\right)$ samples and we make $|\mathcal{H}|$ such calls.

Since the accuracy parameter of the statistical query oracle is $\epsilon/2$, using the triangle inequality, we have that $|\widehat{a}_{\widehat{h}^*} - \min_{h\in\mathcal{H}} a_h| \le \epsilon$.

Finally, the correctness of the algorithm follows from a union bound over the correctness of every call to the oracle. □

7.9 TV Indistinguishability and Replicability

In this section, we will study the connection between TV indistinguishability and replicability.

7.9.1 The Proof of Theorem 7.4.6

We are now ready to establish the connection between TV indistinguishability and replicability. The upcoming result is particularly useful because it provides a *data-independent* way to couple the random variables.

Proof of Theorem 7.4.6. Let $\mathcal{R}$ be Poisson point process with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, where Leb is the Lebesgue measure over $\mathbb{R}_+$ (cf. Theorem 7.7.13, Figure 7.1). The learning rule A' is defined in the following way. For every $S \in (\{\mathcal{X} \times \{0,1\})^n$, let $r = \{(h_i, y_i, t_i)\}_{i\in\mathbb{N}}$ be an infinite sequence of the Poisson point process $\mathcal{R}$ and let $j = \arg\min_{i\in\mathbb{N}}\{t_i : f_S(h_i) > y_i\}$. The output of A' is h_j and we denote it by $A'(S,r)$. We will shortly explain why this is well-defined, except for a measure zero event. The fact that A' is equivalent to A follows from the coupling guarantees of this process (cf. Theorem 7.7.13). In particular, we can instantiate this result with the single random variable $\{A(S)\}$. We can now observe that, except for a measure zero event, (i) since A is absolutely continuous with respect to $\mathcal{P}$, there exists such a density f_S, (ii) the set over which we are taking the minimum is not empty, (iii) the minimum is attained at a unique point. This means that A' is well-defined, except for a measure zero event[8], and, by the correctness of the rejection sampling process (AS19), $A'(S)$ has the desired probability distribution.

[8] Under the measure zero event that at least one of these three conditions does not hold, we let $A'(S,r)$ be some arbitrary classifier.

We now prove that A' is replicable. Since A is ρ-TV indistinguishable, it follows that

$$\mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A(S),A(S'))] \leq \rho.$$

We have shown that A' is equivalent to A, so we can see that $\mathbb{E}_{S,S'\sim\mathcal{D}^n}[d_{\mathrm{TV}}(A'(S),A'(S'))] \leq \rho$. Thus, using the guarantees of Theorem 7.7.13, we have that for any datasets S, S'

$$\mathop{\mathbf{Pr}}_{r\sim\mathcal{R}}[A'(S,r) \neq A'(S',r)] \leq \frac{2d_{\mathrm{TV}}(A'(S),A'(S'))}{1+d_{\mathrm{TV}}(A'(S),A'(S'))}\,.$$

By taking the expectation over S, S', we get that

$$\begin{aligned}\mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}\left[\mathop{\mathbf{Pr}}_{r\sim\mathcal{R}}[A'(S,r) \neq A'(S',r)]\right] &\leq \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}\left[\frac{2d_{\mathrm{TV}}(A'(S),A'(S'))}{1+d_{\mathrm{TV}}(A'(S),A'(S'))}\right]\\ &\leq \frac{2\,\mathbb{E}_{S,S'\sim\mathcal{D}^n}\left[d_{\mathrm{TV}}(A'(S),A'(S'))\right]}{1+\mathbb{E}_{S,S'\sim\mathcal{D}^n}\left[d_{\mathrm{TV}}(A'(S),A'(S'))\right]}\\ &\leq \frac{2\rho}{1+\rho},\end{aligned}$$

where the first inequality follows from Theorem 7.7.13 and taking the expectation over S, S', the second inequality follows from Jensen's inequality, and the third inequality follows from the fact that $f(x) = 2x/(1+x)$ is increasing. Now notice that since the source of randomness $\mathcal{R}$ is independent of S, S', we have that

$$\mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n}\left[\mathop{\mathbf{Pr}}_{r\sim\mathcal{R}}[A'(S,r) \neq A'(S',r)]\right] = \mathop{\mathbf{Pr}}_{S,S'\sim\mathcal{D}^n,r\sim\mathcal{R}}[A'(S,r) \neq A'(S',r)]\,.$$

Thus, we have shown that

$$\mathop{\mathbf{Pr}}_{S,S'\sim\mathcal{D}^n,r\sim\mathcal{R}}[A'(S,r) \neq A'(S',r)] \leq \frac{2\rho}{1+\rho}\,,$$

so the algorithm A' is n-sample $\frac{2\rho}{1+\rho}$-replicable, which concludes the proof. □

7.9.2 A General Equivalence Result

In this section, we focus on the following two stability/replicability definitions.

Definition 7.9.1 (Replicability (ILPS22))**.** *Let $\mathcal{R}$ be a distribution over random strings. A learning rule A is ρ-replicable if for any distribution μ over $\mathcal{I}$ and two independent instances $I, I' \sim \mu$ it holds that*

$$\mathop{\mathbf{Pr}}_{I,I'\sim\mu,r\sim\mathcal{R}}[A(I,r) \neq A(I',r)] \leq \rho\,.$$

Definition 7.9.2 (Total Variation Indistinguishability)**.** *A learning rule A is ρ-*TV *indistinguishable if for any distribution μ and two independent instances $I, I' \sim \mu$ it holds that*

$$\mathop{\mathbb{E}}_{I,I' \sim \mu}[d_{\mathrm{TV}}(A(I), A(I'))] \leq \rho\,.$$

A learning rule A is ρ-fixed prior TV *indistinguishable if for any distribution μ, there exists some prior $\mathcal{P}_\mu$ such that for $I \sim \mu$ it holds that*

$$\mathop{\mathbb{E}}_{I \sim \mu}[d_{\mathrm{TV}}(A(I), \mathcal{P}_\mu)] \leq \rho\,.$$

Definition 7.9.3 (Pseudo-Global Stability)**.** *Let $\mathcal{R}$ be a distribution over random strings. A learning rule A is said to be (η, ν)-pseudo-globally stable if for any distribution μ there exists an element $o_r \in \mathcal{O}$ for every $r \in \operatorname{supp}(\mathcal{R})$ (depending on μ) such that*

$$\mathop{\mathbf{Pr}}_{r \sim \mathcal{R}}\left[\mathop{\mathbf{Pr}}_{I \sim \mu}[A(I, r) = o_r] \geq \eta\right] \geq \nu\,.$$

Our general equivalence result follows.

Proposition 7.9.4 (TV Indistinguishability $\equiv$ Replicability)**.** *Let $\mathcal{I}$ be an input space and $\mathcal{O}$ be an output space.*

- *If a learning rule A is ρ-replicable, then it is also ρ-*TV *indistinguishable.*
- *Consider a prior distribution $\mathcal{P}$ over $\mathcal{O}$. Consider a learning rule A that is ρ-*TV *indistinguishable and absolutely continuous with respect to $\mathcal{P}$. Then, there exists a learning rule A' that is equivalent to A and A' is $2\rho/(1+\rho)$-replicable.*

We remark that one can adapt the proofs of Theorem 7.4.1 and Theorem 7.4.6 by setting $\mathcal{I} = (\mathcal{X} \times \{0,1\})^n$, $\mu = \mathcal{D}^n$ and $\mathcal{O} = \{0,1\}^{\mathcal{X}}$. Moreover, when $\mathcal{I}$ is countable, the design of the reference probability measure works in a similar way. Hence, we get the following corollary.

Corollary 7.9.5. *Let $\mathcal{I}$ be a countable domain and let A be a learning rule that is ρ-*TV *indistinguishable. Then, there exists a $\frac{2\rho}{1+\rho}$-replicable learning rule A' that is equivalent to A.*

7.10 TV Indistinguishability and Differential Privacy

7.10.1 DP Preliminaries

We introduce some standard tools from the DP literature. We start with the Stable Histograms algorithm (KKMN09; BNS16b). Let $\mathcal{X}$ be some domain and let

$S \in \mathcal{X}^n$ be a (multi)set of its elements. We denote by $\mathrm{freq}_S(x) = \frac{1}{n} \cdot |\{i \in [n] : x_i = x\}|$, i.e., the fraction of times that x appears in S. The following result holds. It essentially allows us to privately publish a short list of elements that appear with high frequency in a dataset.

Lemma 7.10.1 (Stable Histograms (KKMN09; BNS16b))**.** *Let $\mathcal{X}$ be some domain. For*

$$n \geq O\left(\frac{\log(1/(\eta\beta\delta))}{\eta\varepsilon}\right)$$

there exists an (ε, δ)-differentially private algorithm `StableHist` *which, with probability at least $1 - \beta$, on input $S = (x_1, \ldots, x_n) \in \mathcal{X}^n$, outputs a list $L \subseteq \mathcal{X}$ and a sequence of estimates $a \in [0,1]^{|L|}$ such that*

- *Every x with $\mathrm{freq}_S(x) \geq \eta$ appears in L.*
- *For every $x \in L$, the estimate a_x satisfies $|a_x - \mathrm{freq}_S(x)| \leq \eta$.*

We also recall the agnostic private learner for finite classes that was proposed in (KLN$^+$11) and is based on the Exponential Mechanism of (MT07).

Lemma 7.10.2 (Generic Private Learner (KLN$^+$11))**.** *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. There is an $(\varepsilon, 0)$-differentially private algorithm* `GenPrivLearner` *which given*

$$n = O\left(\log(|\mathcal{H}|/\beta) \cdot \max\left\{\frac{1}{\varepsilon\alpha}, \frac{1}{\alpha^2}\right\}\right)$$

samples from $\mathcal{D}$, outputs a hypothesis h such that

$$\mathbf{Pr}\left[\mathrm{err}_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} \mathrm{err}_{\mathcal{D}}(h') + \alpha\right] \geq 1 - \beta\,.$$

Finally, we state a result relating weak learners and privacy.

Theorem 7.10.3 (Weakly Accurate Private Learning $\implies$ Finite Littlestone Dimension (ALMM19))**.** *Let $\mathcal{X}$ be some domain and $H \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class with Littlestone dimension $d \in \mathbb{N} \cup \{\infty\}$ and let A be a weakly accurate learning algorithm (i.e., (α, β)-accurate with $\alpha = 1/2 - \gamma, \beta = 1/2 - \gamma$) for H with sample complexity n that satisfies (ε, δ)-differential privacy with $(\varepsilon, \delta) = (0.1, 1/(n^2 \log(n)))$. Then, $n \geq \Omega(\log^\star(d))$.*

In particular any class that is privately weakly-learnable has a finite Littlestone dimension.

We remark that this theorem appears in (ALMM19) with accuracy constant 0.1. However, it is known from (DRV10) that a DP algorithm with error $1/2 - \gamma$ can be boosted to one with arbitrarily small error with negligible loss in the privacy guarantees.

The following result that appears in (BLM20) shows that if $\mathcal{H}$ has finite Littlestone dimension, then there exists a ρ-globally stable learner for this class.

Theorem 7.10.4 (Finite Littlestone Dimension $\implies$ Global Stability (BLM20))**.** *Let $\mathcal{X}$ be some domain and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class with Littlestone dimension $d < \infty$. Let $\alpha > 0$ be the accuracy parameter and define $n = 2^{2^{d+2}+1}4^{d+1} \cdot \left\lceil \frac{2^{d+2}}{\alpha} \right\rceil$. Then, there exists a randomized algorithm $A : (\mathcal{X} \times \{0,1\})^n \times \mathcal{R} \to \{0,1\}^X$ such that for any realizable distribution $\mathcal{D}$ there exists a hypothesis $f_{\mathcal{D}}$ for which*

$$\mathop{\mathbf{Pr}}_{S \sim \mathcal{D}^n, r \sim \mathcal{R}}[A(S,r) = f_{\mathcal{D}}] \geq \frac{1}{(d+1)2^{2^d+1}}, \qquad \mathop{\mathbf{Pr}}_{(x,y) \sim \mathcal{D}}[f_{\mathcal{D}}(x) \neq y] \leq \alpha \,,$$

where $\mathcal{R}$ is the source of internal randomness of A.

We also include a result from (GKM21; BGH$^+$23) which states that replicability implies differential privacy under general input domains[9].

Theorem 7.10.5 (Replicability $\implies$ Differential Privacy (GKM21; BGH$^+$23))**.** *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, where $\mathcal{X}$ is some input domain. If $\mathcal{H}$ is learnable by an n-sample (α, β)-accurate ρ-replicable learner A, for $\alpha \in (0, 1/2), \rho \in (0,1), \beta \in \left(0, \frac{2\rho}{\rho+1} - 2\sqrt{\frac{2\rho}{\rho+1}} + 1\right)$, then, for any $(\alpha', \beta', \epsilon, \delta) \in (0,1)^4$ it is learnable by an $(\alpha + \alpha', \beta')$-accurate (ε, δ)-differentially private learner. Moreover, its sample complexity is*

$$n \cdot \mathrm{poly}(1/\alpha', 1/\varepsilon, \log(1/\delta), \log(1/\beta')) \,.$$

7.10.2 The Proof of Theorem 7.5.1

In this section we show that Global Stability (cf. Definition 7.7.8) implies TV indistinguishability in the context of PAC learning. In particular, we show that given black-box access to a ρ-globally stable learner A whose stable output is α-accurate, e.g., the one described in Theorem 7.10.4, we can transform it to a ρ-TV indistinguishable learner which is $(\alpha + \alpha', \beta)$-accurate, with a multiplicative $\mathrm{poly}(1/\rho, 1/\alpha', \log(1/\beta)$ blow-up in its sample complexity. We remark that this transformation is not restricted to countable domains $\mathcal{X}$. As an intermediate result, we show that global stability implies replicability.

Lemma 7.10.6 (Global Stability $\implies$ Replicability)**.** *Let A be an n-sample ρ-globally stable learner whose stable hypothesis is α-accurate. Then, for every $\rho', \alpha', \beta \in (0,1)^3$, there exists a learner A' (Algorithm 14) that is ρ'-replicable and $(\alpha + \alpha', \beta)$-accurate. Moreover, A' needs*

$$\widetilde{O}\left(\frac{\log(1/\beta)}{\rho'^2 \rho^3}\right)$$

[9] In fact, this result holds for general statistical tasks. The parameters stated in (BGH$^+$23) are slightly looser, but using our boosting results we can generalize them and use the ones that appear in the statement.

oracle calls to A and uses

$$\widetilde{O}\left(\frac{\log(1/\beta)}{\rho^2\rho'^3}\cdot\left(n+\frac{1}{\alpha'^2}\right)\right)$$

samples.

Proof. We first argue about the accuracy and the confidence of the algorithm. Let h_A be the hypothesis such that $\mathbf{Pr}_{S\sim\mathcal{D}^n}[A(S)=h]\geq 1-\rho$. The replicable heavy hitters algorithm (Algorithm 12) guarantees that, with probability at least $1-\beta/2$, h_A will be contained in the output list L (Lemma 7.8.5). We call this event E_0 and we condition on it. In the next step, we call the replicable agnostic learner on L (Algorithm 13). Since there is a hypothesis whose error rate is at most α, we know that the output of the agnostic learner will have error rate at most $\alpha+\alpha'$, with probability at least $1-\beta/2$ (Lemma 7.10.2). Let us call this event E_1. Thus, we see that by taking a union bound over the probabilities of these two events, the error rate of the output of our algorithm will be at most $\alpha+\alpha'$, with probability at least $1-\beta$.

We now shift our focus to the replicability of our algorithm. First, notice that because of the guarantees of the replicable heavy hitters (Lemma 7.8.5) the list L will be the same across two executions when the randomness is shared, with probability at least $1-\rho'/2$. Let us call this event E_2. Similarly, under the event E_2, the output of the agnostic learner will be the same across two executions with probability $1-\rho'/2$. Let us call this event E_3. By taking a union bound over E_2, E_3, we see that the algorithm is ρ'-replicable.

The sample complexity of the algorithm follows by the sample complexity of the replicable heavy hitters and the replicable agnostic learner (Lemma 7.8.5, Lemma 7.10.2). In particular, we need

$$\widetilde{O}\left(n\cdot\frac{\log(1/\beta)}{\rho^2\rho'^3}\right),$$

samples for this step and since the list has size $O(1/\rho')$ we need

$$\widetilde{O}\left(\frac{\log(1/\beta)}{\alpha'^2\rho^2\rho'^3}\right),$$

for the replicable agnostic learner. □

Corollary 7.10.7. *Let A be an n-sample ρ-globally stable learner whose stable hypothesis is α-accurate. Then, for every $\rho', \alpha', \beta \in (0,1)^3$, there exists a learner A' (Algorithm 14) that is ρ'-*TV *indistinguishable and $(\alpha+\alpha',\beta)$-accurate. Moreover, A' needs*

$$\widetilde{O}\left(\frac{\log(1/\beta)}{\rho'^2\rho^3}\right)$$

Algorithm 14 From Global Stability to Replicability

```
Input: Black-box access to a $n$-sample $\rho$-globally stable learner $A$ with $\alpha$-accurate stable hypothesis, sample access to distribution $\mathcal{D}$
Parameters: $\rho', \alpha', \beta \in (0,1)^3$
Output: Classifier $h : \mathcal{X} \to \{0,1\}$
$\mathcal{D}' \leftarrow$ distribution induced by drawing $S \sim \mathcal{D}^n$ and running $A(S)$
$L \leftarrow$ output of ReplicableHeavyHitters (Algorithm 12) with threshold $\rho/2$, error $\rho/4$, confidence $\beta/2$, replicability $\rho'/4$
Output AgnosticReplicableLearner (Algorithm 13) on hypothesis class $L$, with accuracy $\alpha'$, confidence $\beta'/2$ and replicability $\rho'/2$
```

oracle calls to A and uses

$$\widetilde{O}\left(\frac{\log(1/\beta)}{\rho^2 \rho'^3} \cdot \left(n + \frac{1}{\alpha^2}\right)\right)$$

samples.

Proof. The proof follows immediately from Lemma 7.10.6 and the fact that a ρ'-replicable algorithm is ρ'-TV indistinguishable (Theorem 7.4.1). □

We now explain how we can use the previous results in the previous section to design a replicable algorithm for a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ when we know that $\mathcal{H}$ admits a DP learner, for general domains $\mathcal{X}$. Formally, we prove the following result.

Lemma 7.10.8 (Differential Privacy $\implies$ Replicability in General Domains)**.** *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, where $\mathcal{X}$ is some input domain. Let A be an n-sample $(0.1, 1/(n^2 \log(n)))$-differentially private $(1/2-\gamma, 1/2-\gamma)$-accurate learner for $\mathcal{H}$, for some $\gamma \in (0, 1/2]$. Then, for every $\rho, \alpha, \beta \in (0,1)^3$ there exists a learner A' that is ρ-replicable and (α, β)-accurate. Moreover, A' uses*

$$\widetilde{O}\left(\frac{(d+1)^3 2^{3\cdot(2^d+1)} \log(1/\beta)}{\rho^2} \cdot \left(2^{2^{d+2}+1} 4^{d+1} \cdot \left\lceil \frac{2^{d+2}}{\alpha} \right\rceil + \frac{1}{\alpha^2}\right)\right)$$

samples, where d is the Littlestone dimension of $\mathcal{H}$.

Proof. The first step in the proof is to notice that the existence of such a DP learner for $\mathcal{H}$ implies that its Littlestone dimension d is finite ((ALMM19), Theorem 7.10.3). Then, we instantiate Algorithm 14 with the globally stable algorithm from (BLM20) (Theorem 7.10.4) with accuracy $\alpha/2$. Notice that since the random bits for the globally stable need to be different across two executions of the algorithm, we use two different sources of randomness, one that is public, i.e., shared across two executions, and one that is private, i.e., not shared across two executions. Due to Lemma 7.8.3, this is equivalent to the original definition of replicability (Definition 7.1.1). For the remaining two steps, i.e., the replicable heavy-hitters and the replicable agnostic learner, we use public random bits. The sample complexity of the algorithm follows from the sample complexity of Theorem 7.10.4 and Lemma 7.10.6. □

Corollary 7.10.9 (Differential Privacy $\implies$ TV Indistinguishability in General Domains)**.** *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, where $\mathcal{X}$ is some input domain. Let A be an n-sample $(0.1, 1/(n^2 \log(n)))$-differentially private $(1/2-\gamma, 1/2-\gamma)$-accurate learner for $\mathcal{H}$, for some $\gamma \in (0, 1/2]$. Then, for every $\rho, \alpha, \beta \in (0,1)^3$ there exists a learner A' that is ρ-TV indistinguishable and (α, β)-accurate. Moreover, A' uses*

$$\widetilde{O}\left(\frac{(d+1)^3 2^{3\cdot(2^d+1)} \log(1/\beta)}{\rho^2} \cdot \left(2^{2^{d+2}+1} 4^{d+1} \cdot \left\lceil \frac{2^{d+2}}{\alpha} \right\rceil + \frac{1}{\alpha^2}\right)\right)$$

samples, where d is the Littlestone dimension of $\mathcal{H}$.

Proof. The proof of this result follows immediately by Lemma 7.10.8 and the fact that replicable learners are also TV indistinguishable learners (Theorem 7.4.1). □

7.10.3 List-Global Stability $\implies$ TV Indistinguishability

In this section we provide a different TV indistinguishable learner for classes with finite Littlestone dimension that has polynomial sample complexity dependence on the Littlestone dimension of the class. This learner builds upon the results of (GGKM21; GKM21). In particular, (GGKM21) show that a class with finite Littlestone dimension admits a list-globally stable learner (Definition 7.10.10). This learner constructs a sequence of hypothesis classes whose Littlestone dimension is at most that of $\mathcal{H}$, and part of the proof requires that uniform convergence (Definition 7.7.6) holds for all of them. In order to avoid making measurability assumptions on the domain $\mathcal{X}$ and the hypothesis class $\mathcal{H}$ that would imply such a claim, we only state the results for countable $\mathcal{H}$. Nevertheless, we emphasize that they hold for more general settings.

We underline that the result in (GKM21) which designs a pseudo-globally stable learner for classes with finite Littlestone dimension, holds in the setting where $\mathcal{X}$ is finite because it relies on correlated sampling. The reason behind this fact is that they have to convert a DP learner to a pseudo-globally stable one. In our case, we have to show that if $\mathcal{H}$ is learnable by a DP algorithm, it also admits a TV indistinguishable one. The proof of Theorem 7.5.1 follows almost directly from a result appearing in (GKM21).

Definition 7.10.10 (List-Global Stability (GKM21))**.** *A learning algorithm A is said to be m-sample α-accurate (L, η)-list-globally stable if A outputs a set of at* *most L hypotheses and there exists a hypothesis h (that depends on $\mathcal{D}$) such that* $\mathbf{Pr}_{(x_1,y_1),\ldots,(x_m,y_m)\sim\mathcal{D}^n}[h \in A((x_1,y_1),\ldots,(x_m,y_m))] \geq \eta$ *and* $\mathrm{err}_{\mathcal{D}}(h) \leq \alpha$.

(GKM21) showed the following result regarding list m-list-globally stable learners, which is a modification of a result of (GGKM21).

Lemma 7.10.11 (Finite Littlestone ⇒ List-Global Stability (GKM21; GGKM21))**.** *Let* $\alpha, \zeta > 0$. *and* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ *be a countable hypothesis class with* $\mathrm{Ldim}(\mathcal{H}) = d < \infty$*, where* $\mathcal{X}$ *is an arbitrary domain. Then, there is a* $(d\log(1/\zeta)/\alpha)^{O(1)}$*-sample* α*-accurate* $\left(\exp\left((d/\alpha)^{O(1)}\right), \Omega(1/d)\right)$*-list-globally stable learner for* $\mathcal{H}$ *such that, with probability at least* $1-\zeta$*, every hypothesis* h' *in the output list satisfies* $\mathrm{err}_{\mathcal{D}}(h') \leq 2\alpha$.

Algorithm 15 List-Global Stability $\implies$ TV Indistinguishability (Essentially Algorithm 1 in (GKM21))

```
Input:  Black-box access to list-globally stable learner $A$
Parameters:  $\alpha, \beta, \rho, \eta, L$
Output:  Classifier $h : \mathcal{X} \to \{0,1\}$
$\tau \leftarrow 0.5\eta$
$\gamma \leftarrow \frac{10^6 \log(L/(\rho\tau))}{\tau}$
$k_1 \leftarrow \frac{10^6 \log(L/(\rho\tau))}{\tau^2}$
$k_2 \leftarrow \left\lceil \frac{10^6 \gamma^2 \log(L/(\rho\tau))}{\rho^2} \right\rceil$
$m \leftarrow (d\log(k_1/\beta)/\alpha)^{O(1)}$        ▷ Number of samples to run list-globally
    stable learner with parameters $(\alpha, \beta/k_1)$ (Lemma 7.10.11)
for $i \leftarrow 1$ to $k_1$ do
    Draw $S_i \sim \mathcal{D}^m$, run $A$ on $S_i$ to get a set $H_i$
Let $H$ be the set of all $h \in \mathcal{H}$ that appear in at least $\tau \cdot k_1$ of the sets
    $H_1, \ldots, H_{k_1}$
for $j \leftarrow 1$ to $k_2$ do
    Draw $T_j \sim \mathcal{D}^m$, run $A$ on $T_j$ to get a set $G_j$
for $h \in H$ do
    Let $\widehat{Q}_{H,G_1,\ldots,G_{k_2}}(h) = \frac{|\{j\in[k_2] | h \in G_j\}|}{k_2}$
Let $\widehat{\mathcal{P}}_{H,G_1,\ldots,G_{k_2}}$ be the probability distribution on $\mathcal{H}$ defined by
    $$\widehat{\mathcal{P}}_{H,G_1,\ldots,G_{k_2}}(h) = \begin{cases} \frac{\exp(\gamma \widehat{Q}_{H,G_1,\ldots,G_{k_2}}(h))}{\sum_{h'\in H} \exp(\gamma \widehat{Q}_{H,G_1,\ldots,G_{k_2}}(h'))}, & h \in H, \\ 0, & \text{otherwise.} \end{cases}$$
Output $h \sim \widehat{\mathcal{P}}_{H,G_1,\ldots,G_{k_2}}$
```

Proposition 7.10.12 (Adaptation from (GKM21))**.** *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a countable hypothesis class with* $\mathrm{Ldim}(\mathcal{H}) = d < \infty$ *and $\mathcal{X}$ be an arbitrary domain. Then, for all $\alpha, \beta, \rho \in (0,1)^3$, there exists an n-sample ρ-*TV *indistinguishable algorithm (Algorithm 15) that is (α, β)-accurate with respect to the data-generating distribution $\mathcal{D}$, where*

$$n = \mathrm{poly}(d, 1/\alpha, 1/\rho, \log(1/\beta)) .$$

Proof. First, Lemma 7.10.11 guarantees the existence of a list-globally stable learner A for $\mathcal{H}$. We will borrow some notation from (GKM21). We remark that the proof is a simple adaptation of the proof of Theorem 20 in (GKM21) but we include it for completeness. We will use Algorithm 15 essentially appearing in (GKM21) (this algorithm is the same as Algorithm 1 in (GKM21); their algorithm has an additional last step which performs correlated sampling). We will show that Algorithm 15 satisfies the conclusion of Proposition 7.10.12 and is the desired TV indistinguishable learner.

Sample Complexity. The number of samples used by Algorithm 15 is $m \cdot (k_1 + k_2)$, where m is the number of samples used for the black-box list-globally stable learner A. In particular, we have that

$$n(\alpha, \beta, \rho) = \mathrm{poly}(d, 1/\alpha, 1/\rho, \log(1/\beta)) .$$

Accuracy Analysis. By the guarantees of algorithm A, we get that the output of A consists only of hypotheses with distributional error at most α with probability $1 - \beta/k_1$, a union bound implies that this holds for all hypotheses in H with probability $1 - \beta$. This implies the accuracy guarantee for Algorithm 15.

TV Indistinguishability Analysis. Let us set $Q(h) = \mathbf{Pr}_{S \sim \mathcal{D}^n}[h \in A(S)]$, let $H_{\geq 0.9\tau} = \{h \in 2^{\mathcal{X}} : Q(h) \geq 0.9\tau\}$ and $H_{\geq 1.1\tau} = \{h \in 2^{\mathcal{X}} : Q(h) \geq 1.1\tau\}$. First, Algorithm 15 creates the set H that contains all $h \in \mathcal{H}$ that appear in at least $\tau \cdot k_1$ of the realizations $A(S_1), \ldots, A(S_{k_1})$.

The first lemma controls the probability that H contains hypotheses that are "heavy hitters" for A and does not contain hypotheses h whose $Q(h)$ is small.

Lemma 7.10.13 (Adaptation of Lemma 22 in (GKM21))**.** *Let $\mathcal{E}$ denote the good event that $H_{1.1\tau} \subseteq H \subseteq H_{0.9\tau}$. Then $\mathbf{Pr}[\mathcal{E}] \geq 1 - \rho$, where the randomness is over the datasets $S_1, ..., S_{k_1}$ and A.*

Proof. We will first show that $\mathbf{Pr}[H_{1.1\tau} \subseteq H] \geq 1 - \rho/2$. Since A outputs a list of size at most L, $H_{1.1\tau} \leq \frac{L}{1.1\tau} \leq L/\tau$. For any $f \in H_{1.1\tau}$, we have that $1\{f \in H\}$ is an i.i.d. Bernoulli random variable with success probability $Q(f) \geq 1.1\tau$. Hoeffding's inequality implies that

$$\mathbf{Pr}[f \notin H] \leq \exp(-0.02\tau^2 k_1) \leq 0.01\rho\tau/L .$$

A union bound over all hypotheses in $H_{1.1\tau}$, implies the desired inequality. The other direction follows by a similar argument and we refer to (GKM21) for the complete argument. □

The next step is to define the distribution $\mathcal{P}$ with density $\mathcal{P}(h) \propto \exp(\gamma Q(h))1\{h \in H_{\geq 0.9\tau}\}$. We can also define $\mathcal{P}_H(h) \propto \exp(\gamma Q((h))1\{h \in H\}$, where γ is as in Algorithm 15. The next lemma relates the two distributions.

Lemma 7.10.14 (Adaptation of Lemma 23 in (GKM21))**.** *Under the event $\mathcal{E}$, it holds that $d_{\mathrm{TV}}(\mathcal{P}, \mathcal{P}_H) \leq \rho/2$.*

Proof. The proof is exactly the same as the one of Lemma 23 in (GKM21) with the single modification that we pick γ to be of different value, indicated by Algorithm 15. □

Given a list-globally stable learner A (which exists thanks to Lemma 7.10.11), we can construct the distribution over hypotheses $\widehat{\mathcal{P}}_{H,G_1,...,G_{k_2}}$ appearing in Algorithm 15. We can then relate the empirical distribution $\widehat{\mathcal{P}}_{H,G_1,...,G_{k2}}$ with its population analogue $\mathcal{P}_H$.

Lemma 7.10.15 (Adaptation of Lemma 24 in (GKM21))**.** *It holds that $\mathbb{E}[d_{\mathrm{TV}}(\mathcal{P}_H, \widehat{\mathcal{P}}_{H,G_1,...,G_{k2}})] \leq \rho/2$, where the expectation is over the sets $T_1, ..., T_{k_2}$ and the randomness of A.*

Proof. The proof is exactly the same as the one of Lemma 24 in (GKM21) with the single modification that we pick k_2 to be of different value, indicated by Algorithm 15. □

Combining the above lemmas (as in (GKM21)), we immediately get that $\mathbb{E}[d_{\mathrm{TV}}(\mathcal{P}, \widehat{\mathcal{P}}_{H,G_1,...,G_{k_2}})] \leq \rho$, where the expectation is over all the sets $S_1, ..., S_{k_1}$ and $T_1, ..., T_{k_2}$ given as input to the learner A and A's internal randomness. Note that $\mathcal{P}$ is independent of the data and depends only on A. Both $\mathcal{P}$ and $\widehat{\mathcal{P}}_{H,G_1,...,G_{k_2}}$ are supported on a finite domain. Note that Algorithm 15 that, given a training set S, outputs the distribution over hypotheses $\widehat{\mathcal{P}}_{H,G_1,...,G_{k_2}}$ (obtained by Algorithm 15) satisfies TV indistinguishiability with parameter 2ρ using triangle inequality. Hence, two independent runs of Algorithm 15 will be 2ρ-close in total variation in expectation and the algorithm is TV indistinguishable, as promised.

□

7.10.4 The Proof of Theorem 7.5.2

We are now ready to show that TV indistinguishability implies approximate DP. We first start by showing that a non-trivial TV indistinguishable learner for a class $\mathcal{H}$ gives rise to a non-trivial DP learner for $\mathcal{H}$. The algorithm is described in Algorithm 16. The result then follows from the fact that classes which admit non-trivial DP learners have finite Littlestone dimension (Theorem 7.10.3).

Algorithm 16 From TV Indistinguishability to Differential Privacy

```
Input: Black-box access to $(\alpha,\beta)$-accurate $\rho$-TV Indistinguishable Learner $A$, Sample $S$
Parameters: $\alpha', \beta', \varepsilon, \delta$
Output: Classifier $h : \mathcal{X} \to \{0,1\}$
$k \leftarrow O_{\beta,\rho}\left(\frac{\log(\log(1/\beta')/(\beta'\delta))}{\varepsilon}\right), k' \leftarrow O_{\beta,\rho}(\log(1/\beta'))$
Break $S$ into disjoint $\{S_i^j\}_{i\in[k],j\in[k']}$ with $|S_i^j| = n, \forall i \in [k], j \in [k']$
$\mathcal{P} \leftarrow$ data-independent reference probability measure from Claim 19
$(X_1^j, \ldots, X_k^j) \leftarrow \Pi_{\mathcal{R}}(A(S_1^j), \ldots, A(S_k^j)), \forall j \in [k']$ using the Poisson point process $\mathcal{R}$ with intensity $\mathcal{P} \times \text{Leb} \times \text{Leb}$   ▷ $A(S_i^j)$ is a distribution over classifiers, the coupling $\Pi_{\mathcal{R}}$ is described in Theorem 7.7.13.
Compute list $L^j \leftarrow$ StableHist$(X_1^j, \ldots, X_k^j)$, with $\eta = O_{\beta,\rho}(1/\log(1/\beta'))$, correctness $\beta'/3$, privacy $(\varepsilon/2, \delta), \forall j \in [k']$   ▷ Lemma 7.10.1
$\widetilde{L}^j \leftarrow$ Remove elements from $L^j$ that appear less than $\eta/2$ times, $\forall j \in [k']$
Output GenPrivLearner$(\widetilde{L}^1, \ldots, \widetilde{L}^{k'})$ with accuracy $(\alpha'/2, \beta'/3)$, privacy $(\varepsilon/2, 0)$   ▷ Lemma 7.10.2
```

Before we state the result formally, let us first provide some intuition behind the approach. On a high level it resembles the approaches of (BLM20; GKM21; BGH$^+$23) to show that (pseudo-)global stability implies differential privacy. We consider k' different batches of k datasets of size n. For each such batch, our goal is to couple the k different executions of the algorithm on an input of size n, so that most of these outputs are, with high probability, the same. One first approach would be to use a random variable as a "pivot" element in each batch: we first draw $A(S_1^1)$ according to its distribution and the remaining $\{A(S_i^1)\}_{i\in[k]\setminus\{1\}}$ from their optimal coupling with $A(S_1^1)$, given its realized value. Even though this coupling has the property that, in expectation, most of the outputs will be the same, it is not robust at all. If the adversary changes a point of S_1^1, then the values of all the outputs will change! This is not privacy preserving. For this reason, we use the coupling that is described in Theorem 7.7.13. We use the fact that $\mathcal{X}$ is countable to design a reference probability measure $\mathcal{P}$ that is independent of the data. This is the key step that leads to privacy-preservation. Then, we can argue that if we follow this approach for multiple batches, there will be a classifier whose frequency and performance are non-trivial. The next step is to feed all these hypotheses into the Stable Histograms algorithm (cf. Lemma 7.10.1), which will output a list of frequent hypotheses that includes the non-trivial one we mentioned above. Finally, we feed these hypotheses into the Generic Private Learner (cf. Lemma 7.10.2) and we get the desired result.

Proposition 7.10.16. *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ where $\mathcal{X}$ is countable. Assume that $\mathcal{H}$*

is learnable by an (α, β)-accurate ρ-TV indistinguishable learner A using n_{TV} samples, where $\rho \in (0,1), \alpha \in (0,1/2), \beta \in (0,(1-\rho)/(1+\rho))$. Then, for any $(\alpha', \beta', \varepsilon, \delta) \in (0,1)^4$, it is also learnable by an $(\alpha+\alpha', \beta')$-accurate (ε, δ)-differentially private learner and the sample complexity is

$$n_{\mathrm{DP}} = O_{\beta,\rho}\left(\frac{\log(1/\beta') \cdot \log(\log(1/\beta')/(\beta'\delta))}{\varepsilon} + \log(1/\eta\beta') \cdot \max\left\{\frac{1}{\varepsilon\alpha'}, \frac{1}{\alpha'^2}\right\}\right) \cdot n_{\mathrm{TV}} \, .$$

Proof. Let A be the TV indistinguishable algorithm. We need to argue that the output of Algorithm 16 is $(\alpha+\alpha', \beta')$-accurate and (ε, δ)-DP. We start with the former property.

Performance Guarantee. Let us consider the following experiment. We draw k samples, each one of size $n = n_{\mathrm{TV}}$. Let $S_1^1, \ldots, S_k^1$ be these samples and $A(S_1^1), \ldots, A(S_k^1)$ be the distributions of the outputs of the algorithm on these samples. We denote by X_i^1 the random variable that follows the distribution $A(S_i^1)$. Let us consider a coupling of this collection of variables. Then, we have that

$$\begin{aligned}\mathop{\mathbb{E}}_{\text{coupling}}\left[\min_{j\in[k]} \sum_{i=1}^{k} \mathbb{1}_{X_i^1 \neq X_j^1}\right] &\leq \mathop{\mathbb{E}}_{\text{coupling}}\left[\sum_{i=1}^{k} \mathbb{1}_{X_i^1 \neq X_1^1}\right] \\ &= \sum_{i=1}^{k} \mathop{\mathbb{E}}_{\text{coupling}}\left[\mathbb{1}_{X_i^1 \neq X_1^1}\right] \\ &= \sum_{i=1}^{k} \mathop{\mathbf{Pr}}_{\text{coupling}}[X_i^1 \neq X_1^1] \, .\end{aligned}$$

Note that the above hold for any coupling between the random variables $(X_i^1)_{i\in[n]}$. Let us fix the DP parameters (ϵ, δ). We will use the coupling protocol of Theorem 7.7.13 with $\Omega = \{0,1\}^{\mathcal{X}}$, $\mathcal{P}$ the probability measure described in Claim 19, and $\mathcal{R}$ the Poisson point process with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$. We remark that this choice of $\mathcal{P}$ satisfies two properties: the collection $A(S_i^1)$ is absolutely continuous with respect to $\mathcal{P}$ and $\mathcal{P}$ is data-independent, so it will help us establish the differential privacy guarantees. The guarantees of the coupling of Theorem 7.7.13 imply that

$$\mathop{\mathbf{Pr}}_{\mathcal{R}}[X_i^1 \neq X_1^1] \leq \frac{2d_{\mathrm{TV}}(X_i^1, X_1^1)}{1 + d_{\mathrm{TV}}(X_i^1, X_1^1)} \, ,$$

for all $i \in [k]$. Thus, we have that

$$\mathop{\mathbb{E}}_{\mathcal{R}}\left[\min_{j\in[k]} \sum_{i=1}^{k} \mathbb{1}_{X_i^1 \neq X_j^1}\right] \leq \sum_{i=1}^{k} \frac{2d_{\mathrm{TV}}(X_i^1, X_1^1)}{1 + d_{\mathrm{TV}}(X_i^1, X_1^1)} \, .$$

By taking the expectation over the random draws of the samples $S_1, \ldots, S_k$, we see that

$$\begin{aligned}
\mathop{\mathbb{E}}_{S_1^1,\ldots,S_k^1,\mathcal{R}}\left[\sum_{i=1}^{k} \mathbb{1}_{X_i^1 \neq X_1^1}\right] &\leq \mathop{\mathbb{E}}_{S_1^1,\ldots,S_k^1}\left[\sum_{i=1}^{k} \frac{2d_{\mathrm{TV}}(X_i^1, X_1^1)}{1+d_{\mathrm{TV}}(X_i^1, X_1^1)}\right] \\
&= \sum_{i=1}^{k} \mathop{\mathbb{E}}_{S_1^1,\ldots,S_k^1}\left[\frac{2d_{\mathrm{TV}}(X_i^1, X_1^1)}{1+d_{\mathrm{TV}}(X_i^1, X_1^1)}\right] \\
&\leq \sum_{i=1}^{k} \frac{2\,\mathbb{E}_{S_1^1,\ldots,S_k^1}[d_{\mathrm{TV}}(X_i^1, X_1^1)]}{1+\mathbb{E}_{S_1^1,\ldots,S_k^1}[d_{\mathrm{TV}}(X_i^1, X_1^1)]} \\
&\leq \frac{2\rho}{1+\rho} \cdot k,
\end{aligned}$$

where the second to last step follows by Jensen's inequality since the function $f(x) = 2x/(1+x)$ is concave in $(0,1)$ and the last step because $\mathbb{E}_{S_1^1,\ldots,S_k^1}[d_{\mathrm{TV}}(X_i^1, X_1^1)] \leq \rho$ and f is increasing in $(0,1)$. To make the notation cleaner, we let $\rho' = \frac{2\rho}{1+\rho}$. Notice that if $\rho < 1$ then $\rho' < 1$. Now using Markov's inequality we get that

$$\mathbf{Pr}\left[\sum_{i=1}^{k} \mathbb{1}_{X_i^1 \neq X_1^1} \geq \nu k \rho'\right] \leq \frac{1}{\nu} \implies \mathbf{Pr}\left[\sum_{i=1}^{k} \mathbb{1}_{X_i^1 = X_1^1} \geq (1-\nu\rho')k\right] \geq 1 - \frac{1}{\nu},$$

where the probability is with respect to the randomness of the samples and the coupling.

We denote by $\mathcal{E}_\nu^1 = \left\{\sum_{i=1}^{k} \mathbb{1}_{X_i^1 = X_1^1} \geq (1-\nu\rho')k\right\}$ the event that a $(1-\nu\rho')$-fraction of the outputs has the same value. Let us now focus on the number of classifiers in a single experiment that are correct, i.e., their error rate is at most $\alpha < 1/2$. Let $Y_i^1 = \mathbb{1}_{\mathrm{err}(X_i^1) \geq 1/2}$. Notice that because of the coupling we have used, $\{Y_i^1\}_{i=1}^k$ are not independent, so we cannot simply apply a Chernoff bound to get concentration. Let $\mathcal{E}_\beta^1$ be the event that the classifier X_1^1 is correct. We know that $\mathbf{Pr}[\mathcal{E}_\beta^1] \geq 1-\beta$, where the probability is taken with respect to the random draws of the input and the randomness of the algorithm. Now notice that under the event $\mathcal{E}_\nu^1 \cap \mathcal{E}_\beta^1$ at least $(1-\nu\rho')k$ classifiers are correct and have the same output. By a union bound we see that

$$\mathbf{Pr}[\mathcal{E}_\nu^1 \cap \mathcal{E}_\beta^1] \geq 1 - \beta - \frac{1}{\nu}.$$

We now pick ν so that

$$1 - \beta - \frac{1}{\nu} = \frac{1-\beta-\rho'}{2} > 0 \implies \nu = \frac{2}{\rho' - \beta + 1}.$$

Thus, under $\mathcal{E}_\nu^1 \cap \mathcal{E}_\beta^1$ there are $\frac{1-\beta-\rho'}{1-\beta+\rho'}k$ classifiers that are equal to one another and are correct. We let $q = \frac{1-\beta-\rho'}{1-\beta+\rho'}$. As we discussed, the probability of this event is at least $\frac{1-\beta-\rho'}{2} = p$, so if we execute it k' times we have that with probability at least $1 - e^{-pk'}$ it will occur at least once, i.e., $\mathbf{Pr}\left[\cup_{j\in[k']}\{\mathcal{E}^j_\nu \cap \mathcal{E}^j_\beta\}\right] \geq 1 - e^{-pk'}$. We pick $k' = 1/p \cdot \log(3/\beta')$. Thus, with probability at least $1 - \beta'/3$ there is a correct classifier that appears at least qk times. We condition on this event for the rest of proof and we let $S^j_i, X^j_i \sim A(S^j_i)$ be the i-th sample, classifier of the j-th batch, respectively.

The next step is to feed these classifiers into the Stable Histograms algorithm (cf. Lemma 7.10.1). We have shown that there exists a good classifier whose frequency is at least $\eta = \frac{qk}{k\cdot k'} = \frac{q}{k'}$. Thus, our goal is to detect hypotheses with frequency at least $\eta/2$. We pick the correctness parameter of the algorithm to be $\beta'/3$ and the DP parameters to be $(\varepsilon/2, \delta)$. In total, we need

$$n' = O\left(\frac{\log(1/(\eta\beta'\delta))}{\eta\varepsilon}\right) = O\left(\frac{\log(1/\beta')\cdot\log\left(\log(1/\beta')/(qp\beta'\delta)\right)}{qp\varepsilon}\right),$$

hypotheses in our list. Since $n' = k \cdot k'$ it suffices to pick

$$k = O\left(\frac{\log\left(\log(1/\beta')/(qp\beta'\delta)\right)}{q\varepsilon}\right).$$

Hence, with probability at least $1 - \beta'/3$, the output of the algorithm will be a list L that contains all the hypotheses with frequency at least $\eta/2$ along with estimates a_x such that $|a_x - \mathrm{freq}_S(x)| \leq \eta/2$. Let x^* be the correct and frequent hypothesis whose existence we have established. We know that $a_{x^*} \geq \eta/2$. Since this algorithm is DP, we can drop from its output all the elements $x \in L$ for which $a_x < \eta/2$ without affecting the privacy guarantees. Thus, we end up with a new list L' whose size is $O(1/\eta)$.

The last step of the algorithm is to feed this list into the Generic Private Learner (cf. Lemma 7.10.2) with privacy parameters $(\varepsilon/2, 0)$ and accuracy parameters $(\alpha'/2, \beta'/3)$. The total number of samples we need for this step is

$$n'' = O\left(\log(1/\eta\beta')\cdot\max\left\{\frac{1}{\varepsilon\alpha'}, \frac{1}{\alpha'^2}\right\}\right).$$

Since there is an element in the list whose error is at most α, the guarantees of the algorithm give us that with probability at least $1 - \beta'/3$ the output has error at most $\alpha + \alpha'$.

Thus, by taking a union bound over the correctness of the three steps we described, we see that with probability $1 - \beta'$ the algorithm outputs a hypothesis whose error is at most $\alpha + \alpha'$. We now argue that the algorithm is $(\varepsilon, \delta)-$DP.

Privacy Guarantee. First we need to show that the coupling step is differentially private. This is a direct consequence of the coupling protocol that we have provided (cf. Theorem 7.7.13) and the fact that the reference probability measure is data-independent. If the adversary changes an element in $S^j_i, i \in [k], j \in [k']$, then the coupling is robust, in the sense that if we fix the internal randomness, then at most one of the elements that the coupling outputs will change. The result for the privacy preservation of this step follows by integrating over the internal randomness.

For the remaining two steps, i.e., the Stable Histograms and the Exponential Mechanism the privacy guarantee follows from their definition. Using the privacy composition, we get that overall our algorithm is $(\varepsilon/2, \delta) + (\varepsilon/2, 0) = (\varepsilon, \delta)$-differentially private.

□

Corollary 7.10.17. *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, where $\mathcal{X}$ is a countable domain. If $\mathcal{H}$ is learnable by a (α, β)-accurate ρ-*TV *indistinguishable learner using n_{TV} samples, where $\rho \in (0,1), \alpha \in (0, 1/2), \beta \in (0, (1-\rho)/(1+\rho))$, then* $\mathrm{Ldim}(\mathcal{H}) < \infty$.

Proof. The proof follows directly by combining Proposition 7.10.16 and Theorem 7.10.3. □

7.10.5 Going Beyond Countable $\mathcal{X}$

We now propose an approach that we believe can lead to a generalization of the algorithm beyond countable domains. The only change that we make in the algorithm has to do with Line 6, where for every batch j we pick $\mathcal{P}_j = \frac{1}{k}\sum_{i=1}^{k} A(S_i^j)$. Notice that for every $j \in [k']$ the $\{A(S_i^j)\}_{i \in [k]}$ are absolutely continuous with respect to $\mathcal{P}_j$. However, it is not immediate now that the choice of $\{\mathcal{P}_j\}_{j \in [k']}$ leads to a DP algorithm. We believe that it is indeed the case that the algorithm is approximately differentially private and we leave it as in interesting open problem.

7.11 Amplification and Boosting

7.11.1 The Proof of Theorem 7.6.2

Let us first restate the theorem along with the sample complexity of the algorithm.

Theorem (Indistinguishability Amplification)**.** *Let $\mathcal{P}$ be a reference probability measure over $\{0,1\}^{\mathcal{X}}$ and $\mathcal{D}$ be a distribution over inputs. Consider the source of randomness $\mathcal{R}$ to be a Poisson point process with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, where* Leb *is the Lebesgue measure over $\mathbb{R}_+$. Consider a weak learning rule A that is (i) ρ-*TV *indistinguishable with respect to $\mathcal{D}$ for some $\rho \in (0,1)$, (ii) (α, β)-accurate for $\mathcal{D}$ for $(\alpha, \beta) \in (0,1)^2, \beta < \frac{2\rho}{\rho+1} - 2\sqrt{\frac{2\rho}{\rho+1}} + 1$, and, (iii) absolutely continuous with respect to $\mathcal{P}$ on inputs from $\mathcal{D}$. Then, for any $\rho', \epsilon, \beta' \in (0,1)^3$, there exists an algorithm $\texttt{IndistAmpl}(A, \mathcal{R}, \beta', \epsilon, \rho')$ (Algorithm 17) that is ρ'-*TV *indistinguishable with respect to $\mathcal{D}$ and $(\alpha + \epsilon, \beta')$-accurate for $\mathcal{D}$.*

Let $n_A(\alpha, \beta, \rho)$ denote the sample complexity of the weak learning rule A with input β', ϵ, ρ'. Then, the learning rule `IndistAmpl`$(A, \mathcal{R}, \beta', \epsilon, \rho')$ *uses*

$$\widetilde{O}\left(\frac{\log^3\left(\frac{1}{\beta'}\right)}{\left(\frac{2\rho}{\rho+1}-2\sqrt{\frac{2\rho}{\rho+1}}+1-\beta\right)^2\left(1-\sqrt{\frac{2\rho}{\rho+1}}\right)\epsilon^2\rho'^2}\cdot n_A(\alpha,\beta,\rho)\right)$$

i.i.d. samples from $\mathcal{D}$.

Algorithm 17 Amplification of Indistinguishability Guarantees

```
Input: Black-box access to $(\alpha, \beta)$-accurate $\rho$-TV Indistinguishable Learner $A$, Sample access to $\mathcal{D}$, Access to Poisson point process $\mathcal{R}$ with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$   ▷ $\mathcal{P}$ is the reference probability measure from Claim 19.
Parameters: $\beta', \epsilon, \rho'$
Output: Classifier $h : \mathcal{X} \to \{0, 1\}$
$\eta, \nu \leftarrow \sqrt{\frac{2\rho}{1+\rho}}, \sqrt{\frac{2\rho}{1+\rho}}$
$\mathcal{P} \leftarrow$ data-independent reference probability measure from Claim 19
$k \leftarrow \frac{\log(3/\beta')}{1-\nu-\beta/(1-\eta)}$
$r_i \leftarrow$ an infinite sequence of the Poisson Point Process $\mathcal{R}$, $\forall i \in [k]$   ▷ cf. Theorem 7.7.13.
$\mathcal{D}_{r_i} \leftarrow$ the distribution of hypotheses that is induced by $A(S, r_i)$ when $S \sim \mathcal{D}^n, \forall i \in [k]$
$L_i \leftarrow \mathrm{HeavyHitters}\left(\mathcal{D}_{r_i}, \frac{3}{4}(1-\eta), \frac{1}{4}(1-\eta), \rho'/(2k), \beta'/(3k)\right), \forall i \in [k]$   ▷ Algorithm 12.
$\left(\widehat{h}_i, \widehat{\mathrm{err}}(\widehat{h}_i)\right) \leftarrow \mathrm{AgnosticLearner}(L_i, \epsilon/2, \rho'/(2k), \beta'/(3k)), \forall i \in [k]$   ▷ Algorithm 13.
for $i \leftarrow 1$ to $k$ do
    if $\widehat{\mathrm{err}}(\widehat{h}_i) \leq \alpha + \epsilon/2$ then
        Output $\widehat{h}_i$
Output the all 1 classifier
```

Proof. Since A is ρ-TV indistinguishable there is an equivalent learning rule A' that is $\frac{2\rho}{1+\rho}$-replicable (cf. Theorem 7.4.6) and uses randomness $\mathcal{R}$, where $\mathcal{R}$ is a Poisson point process with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, with Leb being the Lebesgue measure over $\mathbb{R}_+$. Let

$$\mathcal{R}_\eta = \left\{ r \in \mathcal{R} : \exists h \in \mathcal{H} \text{ s.t. } \Pr_{S \sim \mathcal{D}^n}[A'(S, r) = h] \geq 1 - \eta \right\},$$

We have that $\mathbf{Pr}_{r\sim\mathcal{R}}[r \in \mathcal{R}_\eta] \geq 1-\nu$, for $\eta = \frac{\frac{2\rho}{1+\rho}}{\nu}, \nu \in \left[\frac{2\rho}{1+\rho}, 1\right)$ (cf. Claim 20). For each $r \in \mathcal{R}_\eta$ let $h_r \in \mathcal{H}$ be an element that witnesses its inclusion in $\mathcal{R}_\eta$[10]. Notice that since A' is (α, β)-accurate there is at most a $\frac{\beta}{1-\eta}$-fraction of $r \in \mathcal{R}$ such that $r \in \mathcal{R}_\eta, \mathrm{err}(h_r) > \alpha$. Let $\mathcal{R}^*_\eta = \{r \in \mathcal{R}_\eta : \mathrm{err}(h_r) \leq \alpha\}$. Now notice that $\mathbf{Pr}_{r\sim\mathcal{R}}[r \in \mathcal{R}^*_\eta] \geq 1-\nu-\frac{\beta}{1-\eta}$. Thus, by picking $k = \frac{\log(3/\beta')}{1-\nu-\beta/(1-\eta)}$ i.i.d. samples from $\mathcal{R}$ we have that with probability at least $1-\beta'/3$ there will be some $r_{i^*} \in \mathcal{R}^*_\eta$. We denote this event by $\mathcal{E}_1$ and we condition on it for the rest of the proof.

Let us now focus on the call to the replicable heavy hitters subroutine. We have that, with probability at least $1-\beta'/(3k)$, every call will return a list that contains all the $(1-\eta)$-heavy-hitters and no elements whose mass is less than $(1-\eta)/2$. By a union bound, this happens with probability at least $1-\beta'/3$ for all the calls. Let us call this event $\mathcal{E}_2$ and condition on it for the rest of the proof. Notice that under these two events, the list L_{i^*} that corresponds to r_{i^*} will be non-empty and will contain a classifier whose error is at most α.

We now consider the calls to the replicable agnostic learner. Notice that every list that this algorithm takes as input has size at most $\frac{2}{1-\eta}$. Moreover, with probability at least $1-\beta/3'$, the estimated error of every classifier will be at most $\epsilon/2$ away from its true error. We call this event $\mathcal{E}_3$ and condition on it. Hence, for any $\widehat{h}_j, j \in [k]$, that passes the test in the "if" statement, we have that $\mathrm{err}(\widehat{h}_j) \leq \alpha+\epsilon$. In particular, the call to L_{i^*} will return $\widehat{h}_{i^*}$, with estimated error $\widehat{\mathrm{err}}(\widehat{h}^*_i) \leq \alpha+\epsilon/2$, which means that $\mathrm{err}(\widehat{h}^*_i) \leq \alpha+\epsilon$. Hence, the algorithm will such a classifier and, by a union bound, the total probability that this event happens is at least $1-\beta'$.

The replicability of the algorithm follows from a union bound over the replicability of the calls to the heavy hitters and the agnostic learner (cf. Lemma 7.8.5, Claim 21). In particular, since we call the replicable heavy hitters algorithm k times with replicability parameter $\rho'/(2k)$ and the replicable agnostic leaner k times with replicability parameter $\rho'/(2k)$, we know that with probability at least $1-\rho'$ all these calls will return the same output across two executions of the algorithm.

For the sample complexity notice that each call to the replicable heavy hitters algorithm requires $O\left(\frac{k^2\log(k/\beta'(1-\eta))}{(1-\eta)^3\rho'^2}\right)$ (cf. Lemma 7.8.5.) Under the events we have conditioned on, we see that $|L_i| = O(1/(1-\eta)), \forall i \in [k]$, hence each call to the agnostic learner requires $O\left(\frac{k^2}{(1-\eta)^3\epsilon^2\rho'^2}\log\left(\frac{k(1-\eta)}{\beta'}\right)\right)$ (cf. Claim 21). Substituting the value of k gives us that the sample complexity is at most

$$O\left(\frac{\log^3\left(\frac{\log(1/\beta')}{\beta'((1-\eta)(1-\nu)-\beta)}\right)}{((1-\eta)(1-\nu)-\beta)^2(1-\eta)\epsilon^2\rho'^2}\right).$$

Plugging in the values of η, ν we get the stated bound. □

[10] If there are multiple such elements then we pick an arbitrary one using a consistent rule.

7.11.2 The Proof of Theorem 7.6.3

Let us first recall the result we need to prove along with its sample complexity.

Theorem (Accuracy Boosting)**.** *Let $\mathcal{P}$ be a reference probability measure over $\{0,1\}^{\mathcal{X}}$ and $\mathcal{D}$ be a distribution over inputs. Consider the source of randomness $\mathcal{R}$ to be a Poisson point process with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, where* Leb *is the Lebesgue measure over $\mathbb{R}_+$. Consider a weak learning rule A that is (i) ρ-*TV *indistinguishable with respect to $\mathcal{D}$ for some $\rho \in (0,1)$, (ii) $(1/2-\gamma,\beta)$-accurate for $\mathcal{D}$ for some $\gamma \in (0,1/2), \beta \in \left(0, \frac{2\rho}{\rho+1} - 2\sqrt{\frac{2\rho}{\rho+1}} + 1\right)$, and, (iii) absolutely continuous with respect to $\mathcal{P}$ on inputs from $\mathcal{D}$. Then, for any $\rho', \epsilon, \beta' \in (0,1)^3$, there exists an algorithm $\texttt{IndistBoost}(A,\mathcal{R},\epsilon)$ (Algorithm 18) that is ρ'-*TV *indistinguishable with respect to $\mathcal{D}$ and (ϵ,β')-accurate for $\mathcal{D}$.*

If $n_A(\gamma,\beta,\rho)$ is the sample complexity of the weak learning rule A with input γ,β,ρ, then $\texttt{IndistBoost}(A,\mathcal{R},\epsilon)$ uses

$$\widetilde{O}\left(\frac{n_A(\gamma,\beta'\epsilon\gamma^2/6,\rho\epsilon\gamma^2/(3(1+\rho)))\log(1/\beta')}{\epsilon^2\gamma^2} + \frac{\log(1/\beta')}{(2\rho/(1+\rho))^2\epsilon^3\gamma^2}\right)$$

i.i.d. samples from $\mathcal{D}$.

Proof of Theorem 7.6.3. In the sample complexity bound of Theorem 7.6.3, we remark that the first term is the number of samples used by the `RejectionSampling` mechanism (appearing in (ILPS22)) in the T rounds and the second term controls the number of samples used for the `IndistingTestMeasure` procedure (appearing in (ILPS22)) for the T rounds (see Algorithm 18). Let $[T] = \{1,...,T\}$. As in (ILPS22)[Theorem 6.1], we consider that the shared randomness between the two executions consists of a collection of $3T$ tapes with uniformly random bits. We denote the j-th tape in round t by $\mathcal{R}_t^{(j)}$ for $j \in [3]$ and $t \in [T]$. Since A is n-sample ρ-TV indistinguishable there is an equivalent learning rule A' that is n-sample $\frac{2\rho}{1+\rho}$-replicable (cf. Theorem 7.4.6) and uses randomness $\mathcal{R}$, where $\mathcal{R}$ is a Poisson point process with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, with Leb being the Lebesgue measure over $\mathbb{R}_+$. Let us set $\rho' = 2\rho/(1+\rho)$. The boosting algorithm that we provide below interprets the random strings as follows: for any $t \in [T]$, we set $\mathcal{R}_t^{(2)} = \mathcal{R}$ (these will be the tapes used by the equivalent learning algorithm A') and the remaining tapes $\mathcal{R}_t^{(j)}$ corresponds to random samples from the uniform distribution in $[0,1]$ for $j \in \{1,3\}$ (these will be the tapes used by our sub-routines `RejectionSampling` and `IndistingTestMeasure`.

The boosting algorithm works as follows:

1. As in (Ser03), it uses a measure μ_t to assign different scores to points of $\mathcal{X}$. First, $\mu_1(x) = 1$ for any point. We will not delve into the details on how this step works. For details we refer to (Ser03) (as in (ILPS22) since this step is not crucial for the proof).

Algorithm 18 Boosting of Accuracy Guarantee

```
Input: Black-box access to weak $(\frac{1}{2} - \gamma, \beta)$-accurate $\rho$-TV Indistinguishable Learner $A$, Sample $S \sim \mathcal{D}^n$, Access to Poisson point process $\mathcal{R}$ with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$    ▷ $\mathcal{P}$ is the reference probability measure from Claim 19.
Target : $\epsilon, \beta'$
Output: Classifier $h : \mathcal{X} \to \{0, 1\}$
IndistBoost()    ▷ This algorithm appears in (ILPS22)
$\rho' = 2\rho/(1+\rho)$
$T = 100/(\epsilon\gamma^2)$
$\mu_1(x) = 1$
$n_w = n_A\left(\gamma, \frac{\beta'}{3T}, \frac{\rho'}{6T}\right)$
for $t = 1..T$ do
    $\mathcal{D}_{\mu_t}(x) = \frac{\mu_t(x)\mathcal{D}_X(x)}{d(\mu_t)}$
    $S_t \leftarrow n_w/\epsilon \cdot \log(T/\beta')$
    $S_t' \leftarrow$ RejectionSampling $\left(S_t, n_w, \mu_t, \mathcal{R}_t^{(1)}\right)$
    $h_t \sim A\left(S_t', \mathcal{R}_t^{(2)}\right)$
    Update $\mu_{t+1}(x)$ using smooth boosting trick of (Ser03).
    Draw $S_t'' = O(1/(\rho'^2\epsilon^3\gamma^2))$ i.i.d. samples from $\mathcal{D}$
     If IndistingTestMeasure $\left(\mu_{t+1}, S_t'', \mathcal{R}_t^{(3)}, \rho'/(3T), \beta'/(3T)\right) \leq 2\epsilon/3$ then output $\mathrm{sgn}\left(\sum_i h_i\right)$
RejectionSampling($S_{\mathrm{in}}$, size_out, $\mu$, $\mathcal{R}$)
$S_{\mathrm{out}} = \emptyset$
for $(x, y) \in S_{\mathrm{in}}$ do
    Pick $b \in [0, 1]$ using $\mathcal{R}$
    If $\mu(x) \geq b$ then $S_{\mathrm{out}} \leftarrow \mathrm{append}(S_{\mathrm{out}}, (x, y))$
    If $|S_{\mathrm{out}}| >$ size_out then output $S_{\mathrm{out}}$
IndistingTestMeasure($\mu, S, \mathcal{R}, \rho', \beta$)
Call Algorithm 1 in (ILPS22) (see Theorem 7.8.2) with source of randomness $\mathcal{R}$ and dataset $S$, error $\epsilon/3$, confidence $\beta$, replicability $\rho$ and query function $\mu$
```

2. At every round t, the algorithm performs rejection sampling on a fresh dataset S_t using the routine RejectionSampling. This algorithm is TV indistinguishable since it uses the source of randomness $\mathcal{R}_t^{(1)}$ that provides uniform samples in $[0, 1]$ (it is actually replicable).

3. The part of the dataset that was accepted from this rejection sampling pro-

cess is given to replicable learner A', which is equivalent to the TV indistinguishable weak learner A. This algorithm uses the shared Poisson point process $\mathcal{R}_t^{(2)}$ with intensity $\mathcal{P} \times \mathrm{Leb} \times \mathrm{Leb}$, where $\mathcal{P}$ is the reference probability measure from Claim 19, and outputs the same hypothesis with probability $1 - \rho'/(6T)$.

4. Then we use the smooth update rule of (Ser03) to design the new measure μ_{t+1} for the upcoming iteration. This step is deterministic.

5. Last we check whether the boosting procedure is completed. To this end, we check whether μ_t is in expectation small. This step again uses a uniformly random threshold in $[0,1]$ and so makes use of the source $\mathcal{R}_t^{(3)}$.

The algorithm runs for $T = \frac{C}{\epsilon\gamma^2}$ rounds for some numerical constant $C > 0$. Hence, we will assume access to $3T$ tapes of randomness, T with points from the Poisson point process and $2T$ with uniform draws from $[0,1]$. The correctness of the algorithm follows from (Ser03) and (ILPS22)[Theorem 6.1]. As for the TV indistinguishability, this is implied by the replicability of the whole procedure. We have that the weak learner A' is called T times with TV indistinguishability parameter $\rho'/(6T)$, the rejection sampler is called T times so that it outputs $\perp$ with probability $\rho'/(6T)$ and the indistinguishable measure tester is $\rho'/(3T)$-TV indistinguishable and called T times. A union bound gives the desired result. For further details, we refer to (ILPS22) since the analysis is essentially the same.

For the failure probability β', the algorithm can fail if the rejection sampling algorithm outputs $\perp$, if the weak learner fails, and if the replicable SQ oracle (Theorem 7.8.2) fails. We have that the probability that the rejection sampling gives $\perp$ using $n_w/\epsilon \cdot \log(T/\beta')$ is at most β'/T (which can be considered much smaller than $\rho'/(6T)$). Since each one of the three probabilities are upper bounded by $\beta'/(3T)$, the indistinguishable boosting algorithm succeeds with probability $1 - \beta'$. □

7.11.3 Tight Bound Between β, ρ

As we alluded before, Proposition 7.10.16 shows that if we have a ρ-TV indistinguishable (α, β)-accurate learner with $\rho \in (0,1), \alpha \in (0, 1/2), \beta \in \left(0, \frac{1-\rho}{1+\rho}\right)$, then the class $\mathcal{H}$ has finite Littlestone dimension. The reason we need $\beta \in \left(0, \frac{1-\rho}{1+\rho}\right)$ is because, in expectation over the random draws of the samples and the randomness of the coupling, this is the fraction of the executions of the algorithm that will give the same output. The results of (AS19) show that under certain conditions, if we want to couple k random variables whose pairwise TV distance is at most ρ, then under the pairwise optimal coupling the probability that the realization of a pair of them differs is $\frac{2\rho}{1+\rho}$. However, it is unclear what the implication of this result is in the setting we are interested in.

7.11.4 Beyond Countable $\mathcal{X}$

The barrier to push our approach beyond countable $\mathcal{X}$ is very closely related to the one we explained in the DP section. To be more precise, it is not clear how one can design a data-independent reference probability measure $\mathcal{P}$ when $\mathcal{X}$ is uncountable. Hence, one idea would be to use some *data-dependent* probability measure $\mathcal{P}$. This would affect our algorithm in the following way: instead of first sampling the random Poisson point process sequence independently of the data, we first sample $S_1, \ldots, S_k$ and let the reference probability measure be $\mathcal{P} = \frac{1}{k}\sum_{i=1}^{k} A(S_i)$. The difficult step is to show that this algorithm is TV indistinguishable. When we consider a different execution of the algorithm we let $S_1', ..., S_k'$ be the new samples and $\mathcal{P}' = \frac{1}{k}\sum_{i=1}^{k} A(S_i')$ be the new reference probability measure. A natural approach to establish the TV indistinguishability property of the algorithm is to try to couple $\mathcal{P}, \mathcal{P}'$ and show that under this coupling, the expected TV distance of two executions of the new algorithm is small. We leave this question open for future work.

7.12 TV Indistinguishability and Generalization

Recall that in Proposition 7.2.1 we claimed that the generalization bound can shave the dependence on the VC dimension by paying an overhead in the confidence parameter. A similar result appears in (ILPS22) relating replicability to generalization. We now present its proof.

Proof of Proposition 7.2.1. Let S be a sample from $\mathcal{D}^n$. Since A is ρ-TV indistinguishable, it is also ρ-fixed prior TV indistinguishable and let $\mathcal{P}_{\mathcal{D}}$ be the sample-independent prior. Consider two samples $h_1 \sim A(S)$ and $h_2 \sim \mathcal{P}_{\mathcal{D}}$. We consider the following quantities:

- $\widehat{L}(h_1) = \frac{1}{n}\sum_{(x,y)\in S} \mathbb{1}\{h_1(x) \neq y\}$ is the empirical loss of h_1 in S.
- $\widehat{L}(h_2) = \frac{1}{n}\sum_{(x,y)\in S} \mathbb{1}\{h_2(x) \neq y\}$ is the empirical loss of h_2 in S.
- $L(h_1) = \mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h_1(x) \neq y]$ is the population loss of h_1 with respect to $\mathcal{D}$.

We will show that all these three quantities are close to each other. First, let us consider the space of measurable functions $\mathcal{F} = \{f : \|f\|_\infty \leq 1\}$. We have that

$$d_{\mathrm{TV}}(P, Q) = \sup_{f\in\mathcal{F}} \left| \mathop{\mathbb{E}}_{x\sim P}[f(x)] - \mathop{\mathbb{E}}_{x\sim Q}[f(x)] \right| .$$

This means that the total variation distance between two distributions is essentially the worst case bounded distinguisher f. Since $\widehat{L} : \{0,1\}^{\mathcal{X}} \to [0,1]$, we have that

$$\left| \mathop{\mathbb{E}}_{h_1\sim A(S)}\left[\widehat{L}(h_1)\right] - \mathop{\mathbb{E}}_{h_2\sim\mathcal{P}_{\mathcal{D}}}\left[\widehat{L}(h_2)\right] \right| \leq d_{\mathrm{TV}}(A(S), \mathcal{P}_{\mathcal{D}})\,.$$

Similarly, we get that

$$\left| \mathop{\mathbb{E}}_{h_1 \sim A(S)} [L(h_1)] - \mathop{\mathbb{E}}_{h_2 \sim \mathcal{P}_\mathcal{D}} [L(h_2)] \right| \le d_{\mathrm{TV}}(A(S), \mathcal{P}_\mathcal{D}) \,.$$

Now, since A is ρ-fixed prior TV indistinguishable, using Markov's inequality, we have that $\forall \epsilon_1 > 0$,

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^n} \left[\left| \mathop{\mathbb{E}}_{h_1 \sim A(S)} \left[\widehat{L}(h_1) \right] - \mathop{\mathbb{E}}_{h_2 \sim \mathcal{P}_\mathcal{D}} \left[\widehat{L}(h_2) \right] \right| \right] \le \rho \Rightarrow \mathop{\mathbf{Pr}}_{S \sim \mathcal{D}^n} \left[\left| \mathop{\mathbb{E}}_{h_1 \sim A(S)} \left[\widehat{L}(h_1) \right] - \mathop{\mathbb{E}}_{h_2 \sim \mathcal{P}_\mathcal{D}} \left[\widehat{L}(h_2) \right] \right| > \epsilon_1 \right] \le \frac{\rho}{\epsilon_1} \,.$$

In a similar manner, we get

$$\mathop{\mathbf{Pr}}_{S \sim \mathcal{D}^n} \left[\left| \mathop{\mathbb{E}}_{h_1 \sim A(S)} [L(h_1)] - \mathop{\mathbb{E}}_{h_2 \sim \mathcal{P}_\mathcal{D}} [L(h_2)] \right| > \epsilon_1 \right] \le \frac{\rho}{\epsilon_1}$$

We note that, since $\mathcal{P}_\mathcal{D}$ is sample-independent, we have that the statistic

$$\mathop{\mathbb{E}}_{h_2 \sim \mathcal{P}_\mathcal{D}} [\widehat{L}(h_2)] = \frac{1}{n} \sum_{(x,y) \in S} \mathop{\mathbf{Pr}}_{h_2 \sim \mathcal{P}_\mathcal{D}} [h_2(x) \neq y]$$

is a sum of independent random variables with expectation $\mathbb{E}_{h_2 \sim \mathcal{P}_\mathcal{D}}[L(h_2)]$. We can use standard concentration of independent random variables and get

$$\mathop{\mathbf{Pr}}_{S \sim \mathcal{D}^n} \left[\left| \mathop{\mathbb{E}}_{h_2 \sim \mathcal{P}_\mathcal{D}} \left[\widehat{L}_S(h_2) \right] - \mathop{\mathbb{E}}_{h_2 \sim \mathcal{P}_\mathcal{D}} [L_\mathcal{D}(h_2)] \right| \ge \epsilon_2 \right] \le 2e^{-2n\epsilon_2^2} \,,$$

for any $\epsilon_2 > 0$. This means that

$$\mathop{\mathbf{Pr}}_{S \sim \mathcal{D}^n} \left[\left| \mathop{\mathbb{E}}_{h_1 \sim A(S)} \left[\widehat{L}_S(h_1) \right] - \mathop{\mathbb{E}}_{h_1 \sim A(S)} [L_\mathcal{D}(h_1)] \right| \ge 2\varepsilon_1 + \varepsilon_2 \right] \le 2\rho/\epsilon_1 + 2e^{-2n\epsilon_2^2} \,,$$

so we have that, with probability at least $1 - 4\rho/\epsilon - \delta$,

$$\left| \mathop{\mathbb{E}}_{h_1 \sim A(S)} \left[\widehat{L}_S(h_1) \right] - \mathop{\mathbb{E}}_{h_1 \sim A(S)} [L_\mathcal{D}(h_1)] \right| \le \epsilon + \sqrt{\frac{\log(2/\delta)}{2n}} \,.$$

We note that we obtain the result of Proposition 7.2.1 by taking $\epsilon = \sqrt{\rho}$. □

Bibliography

[ABHU15] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Conference on Learning Theory*, pages 167–190. PMLR, 2015. 27, 124

[ABHZ16] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Conference on Learning Theory*, pages 152–192. PMLR, 2016. 27, 124

[ABL$^+$22] Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private and online learnability are equivalent. *ACM Journal of the ACM (JACM)*, 2022. 47, 188

[ABM10] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT*, pages 41–53. Citeseer, 2010. 160

[ABSV14] Pranjal Awasthi, Avrim Blum, Or Sheffet, and Aravindan Vijayaraghavan. Learning mixtures of ranking models. *arXiv preprint arXiv:1410.8750*, 2014. 40

[ACBF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002. 160

[ACK15a] Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. Adaptive Estimation in Weighted Group Testing. In *Proceedings of the 2015 IEEE International Symposium on Information Theory*, ISIT '15, pages 2116–2120. IEEE Computer Society, 2015. 52

[ACK15b] Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. A Chasm Between Identity and Equivalence Testing with Conditional Queries. In *Approximation, Randomization, and Combinatorial Op-*

timization. Algorithms and Techniques., RANDOM '15, pages 449–466, 2015. 52

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):1–27, 2008. 122, 127, 134

[AD98] Javed A Aslam and Scott E Decatur. General bounds on statistical query learning and pac learning with noise via hypothesis boosting. *Information and Computation*, 141(2):85–118, 1998. 36, 87

[ADK15] Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. Optimal Testing for Properties of Distributions. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, pages 3591–3599, 2015. URL: http://arxiv.org/abs/1507.05952. 33

[AEMM22] Ron Amit, Baruch Epstein, Shay Moran, and Ron Meir. Integral probability metrics pac-bayes bounds. *arXiv preprint arXiv:2207.00614*, 2022. 190, 201

[AG12] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012. 160

[AGM17] Juan A Aledo, José A Gámez, and David Molina. Tackling the supervised label ranking problem by bagging weak learners. *Information Fusion*, 35:38–50, 2017. 39, 124

[AJJ$^+$22] Kwangjun Ahn, Prateek Jain, Ziwei Ji, Satyen Kale, Praneeth Netrapalli, and Gil I Shamir. Reproducibility in optimization: Theoretical framework and limits. *arXiv preprint arXiv:2202.04598*, 2022. 43, 46, 167, 187

[AL88] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988. 87

[ALMM19] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860, 2019. 47, 195, 216, 219

[ALMT17] Jacob D Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via differential privacy. 2017. 188

[Ang88] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988. 124

[AO10] Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010. 163

[APA18] Arpit Agarwal, Prathamesh Patil, and Shivani Agarwal. Accelerated spectral ranking. In *International Conference on Machine Learning*, pages 70–79. PMLR, 2018. 40

[AS19] Omer Angel and Yinon Spinka. Pairwise optimal coupling of multiple random variables. *arXiv preprint arXiv:1903.00632*, 2019. 193, 207, 213, 233

[AYPS11] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011. 160

[Bab50] Babington B. Smith. Discussion of Professor Ross's paper. *Journal of Royal Statistical Society B*, 12:53–56, 1950. 60

[Bak16a] Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604), 2016. 27, 28

[Bak16b] Monya Baker. Reproducibility crisis. *Nature*, 533(26):353–66, 2016. 17, 25, 43

[BC18] Rishiraj Bhattacharyya and Sourav Chakraborty. Property Testing of Joint Distributions using Conditional Samples. *Transactions on Computation Theory*, 10(4):16:1–16:20, 2018. 52

[BCB$^+$12] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. 160

[BCBK12] Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41–1. JMLR Workshop and Conference Proceedings, 2012. 160

[BD15] Shai Ben-David. 2 notes on classes with vapnik-chervonenkis dimension 1. *arXiv preprint arXiv:1507.05307*, 2015. 202

[BDCBL92] Shai Ben-David, Nicolò Cesa-Bianchi, and Philip M Long. Characterizations of learnability for classes of {O,. . . , n}-valued functions. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 333–340, 1992. 155

[BDH+20] Ainesh Bakshi, Ilias Diakonikolas, Samuel B Hopkins, Daniel Kane, Sushrut Karmalkar, and Pravesh K Kothari. Outlier-robust clustering of gaussians and other non-spherical mixtures. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 149–159. IEEE Computer Society, 2020. 87

[BDMN05] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005. 36, 87

[BDNP20] Arnab Bhattacharyya, Rathin Desai, Sai Ganesh Nagarajan, and Ioannis Panageas. Efficient statistics for sparse graphical models from truncated samples. *arXiv preprint arXiv:2006.09735*, 2020. 86

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002. 188

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989. 202

[BF15] Maria Florina Balcan and Vitaly Feldman. Statistical active learning algorithms for noise tolerance and differential privacy. *Algorithmica*, 72(1):282–315, 2015. 36, 87

[BF16] Raef Bassily and Yoav Freund. Typicality-based stability and privacy. *arXiv preprint arXiv:1604.03336*, 2016. 188

[BFFSZ19] Róbert Busa-Fekete, Dimitris Fotakis, Balázs Szörényi, and Manolis Zampetakis. Optimal learning of mallows block model. In *Conference on Learning Theory*, pages 529–532. PMLR, 2019. 40, 62, 63

[BFKV98] Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1):35–52, 1998. 36, 87

[BGH+16] Mohammad Bavarian, Badih Ghazi, Elad Haramaty, Pritish Kamath, Ronald L Rivest, and Madhu Sudan. Optimality of correlated sampling strategies. *arXiv preprint arXiv:1612.01041*, 2016. 206, 207

[BGH$^+$23] Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. *arXiv preprint arXiv:2303.12921*, 2023. 46, 47, 50, 187, 188, 189, 195, 196, 197, 198, 207, 217, 224

[BH20] Maria-Florina Balcan and Nika Haghtalab. Noise in classification., 2020. 124

[BHM$^+$21] Olivier Bousquet, Steve Hanneke, Shay Moran, Ramon Van Handel, and Amir Yehudayoff. A theory of universal learning. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 532–541, 2021. 190

[BHM$^+$22] Olivier Bousquet, Steve Hanneke, Shay Moran, Jonathan Shafer, and Ilya Tolstikhin. Fine-grained distribution-dependent learning curves. *arXiv preprint arXiv:2208.14615*, 2022. 190

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003. 209

[BLM20] Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 389–402. IEEE, 2020. 46, 47, 190, 195, 198, 203, 216, 217, 219, 224

[BM02] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002. 202

[BM09] Mark Braverman and Elchanan Mossel. Sorting from noisy information. *arXiv preprint arXiv:0910.1191*, 2009. 40

[BNS$^+$16a] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1046–1059, 2016. 188, 189

[BNS16b] Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 369–380, 2016. 215, 216

[BR93] Mihir Bellare and Phillip Rogaway. The complexity of approximating a nonlinear program. In *Complexity in numerical optimization*, pages 16–32. World Scientific, 1993. 184

[Bre96] Richard Breen. *Regression models: Censored, sample selected, or truncated data*, volume 111. Sage, 1996. 37, 86

[Bro97] Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997. 206

[BS14] Gilles Blanchard and Clayton Scott. Decontamination of mutually contaminated models. In *Artificial Intelligence and Statistics*, pages 1–9. PMLR, 2014. 87

[BSS$^+$20] Guy Bukchin, Eli Schwartz, Kate Saenko, Ori Shahar, Rogerio Feris, Raja Giryes, and Leonid Karlinsky. Fine-grained angular contrastive learning with coarse labels. *arXiv preprint arXiv:2012.03515*, 2020. 34

[BT52a] R.A. Bradley and M.E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39:324, 1952. 60

[BT52b] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 40, 158

[BZ17] Maria-Florina F Balcan and Hongyang Zhang. Sample and computationally efficient learning algorithms under s-concave distributions. *Advances in Neural Information Processing Systems*, 30, 2017. 27, 41, 122, 124

[Can15] Clément L. Canonne. Big Data on the Rise? - Testing Monotonicity of Distributions. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*, ICALP '15, pages 294–305, 2015. 52

[CBDS13] Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. *Advances in Neural Information Processing Systems*, 26, 2013. 163

[CBF98] Nicolo Cesa-Bianchi and Paul Fischer. Finite-time regret bounds for the multiarmed bandit problem. In *ICML*, volume 98, pages 100–108. Citeseer, 1998. 160

[CCK$^+$19] Clément L Canonne, Xi Chen, Gautam Kamath, Amit Levi, and Erik Waingarten. Random Restrictions of High-Dimensional Distributions and Uniformity Testing with Subcube Conditioning. *CoRR*, abs/1911.07357, 2019. 52

[CDCM18] Zhuo Chen, Ruizhou Ding, Ting-Wu Chin, and Diana Marculescu. Understanding the impact of label granularity on cnn-based image classification. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 895–904. IEEE, 2018. 26, 34

[CDGS20] Yu Cheng, Ilias Diakonikolas, Rong Ge, and Mahdi Soltanolkotabi. High-dimensional robust mean estimation via gradient descent. In *International Conference on Machine Learning*, pages 1768–1778. PMLR, 2020. 87

[CDH10] Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier. Label ranking methods based on the plackett-luce model. In *ICML*, 2010. 39, 124

[CDKS17] Clément L. Canonne, Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Testing Bayesian Networks. In *Proceedings of the 30th Annual Conference on Learning Theory, (COLT)*, pages 370–448, 2017. URL: http://arxiv.org/abs/1612.03156. 33, 58, 59

[CDS20] Clément L Canonne, Anindya De, and Rocco A. Servedio. Learning from satisfying assignments under continuous distributions. In *14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 82–101. SIAM, 2020. 52

[CFGM13] Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the Power of Conditional Samples in Distribution Testing. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 561–580. ACM, 2013. 52

[CGAD22] Maxime Cauchois, Suyash Gupta, Alnur Ali, and John Duchi.

Predictive inference with weak supervision. *arXiv preprint arXiv:2201.08315*, 2022. 88

[CH08] Weiwei Cheng and Eyke Hüllermeier. Instance-based label ranking using the mallows model. In *ECCBR Workshops*, pages 143–157, 2008. 39, 124

[CH12] Weiwei Cheng and Eyke Hüllermeier. Probability estimation for multi-class classification based on label ranking. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 83–98. Springer, 2012. 125

[Cha02] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002. 206

[CHH09] Weiwei Cheng, Jens Hühn, and Eyke Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 161–168, 2009. 39, 124

[CKM$^+$19] Clément L. Canonne, Gautam Kamath, Audra McMillan, Jonathan Ullman, and Lydia Zakynthinou. Private Identity Testing for High-Dimensional Distributions. In *arXiv preprint arXiv:1905.11947*, 2019. URL: http://arxiv.org/abs/1905.11947. 33

[CKMY20] Sitan Chen, Frederic Koehler, Ankur Moitra, and Morris Yau. Classification under misspecification: Halfspaces, generalized linear models, and evolvability. *Advances in Neural Information Processing Systems*, 33:8391–8403, 2020. 27, 41, 124, 209

[CKS18] Stephan Clémençon, Anna Korba, and Eric Sibony. Ranking median regression: Learning to order through local consensus. In *Algorithmic Learning Theory*, pages 212–245. PMLR, 2018. 124

[CLN$^+$16] Rachel Cummings, Katrina Ligett, Kobbi Nissim, Aaron Roth, and Zhiwei Steven Wu. Adaptive learning with robust generalization guarantees. In *Conference on Learning Theory*, pages 772–814. PMLR, 2016. 28, 188

[CODA08] Etienne Côme, Latifa Oukhellou, Thierry Denœux, and Patrice Aknin. Mixture model estimation with soft labels. In *Soft Methods for Handling Variability and Imprecision*, pages 165–174. Springer, 2008. 87

[Coh16] A Clifford Cohen. *Truncated and censored samples: theory and applications*. CRC press, 2016. 37, 86

[CPCP14] Yi-Chen Chen, Vishal M Patel, Rama Chellappa, and P Jonathon Phillips. Ambiguously labeled learning using dictionaries. *IEEE Transactions on Information Forensics and Security*, 9(12):2076–2088, 2014. 87

[CPS13] Ioannis Caragiannis, Ariel D Procaccia, and Nisarg Shah. When do noisy votes reveal the truth? In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 143–160, 2013. 40, 62, 63

[CRB20] Vivien Cabannnes, Alessandro Rudi, and Francis Bach. Structured prediction with partial labelling through the infimum loss. In *International Conference on Machine Learning*, pages 1230–1239. PMLR, 2020. 26, 88

[CRS14] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing equivalence between distributions using conditional samples. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1174–1192. SIAM, 2014. 52

[CRS15] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing probability distributions using conditional samples. *SIAM Journal on Computing*, 44(3):540–616, 2015. 52

[CS12] Jesús Cid-Sueiro. Proper losses for learning from partial labels. *Advances in neural information processing systems*, 25, 2012. 88

[CSGGSR14] Jesús Cid-Sueiro, Darío García-García, and Raúl Santos-Rodríguez. Consistency of losses for learning from weak labels. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 197–210. Springer, 2014. 26, 88

[CSJT09] Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 919–926. IEEE, 2009. 87

[CST11a] Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011. 26, 87, 88

[CST11b] Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011. 87

[CSV17] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from Untrusted Data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 47–60, 2017. 51, 87

[CSZ06] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006. 87

[CV20] Stéphan Clémençon and Robin Vogel. A multiclass classification approach to label ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 1421–1430. PMLR, 2020. URL: https://arxiv.org/abs/2002.09420. 124

[CW01] Anthony Carbery and James Wright. Distributional and L^q norm inequalities for polynomials over convex bodies in $\mathbb{R}^n$. *Mathematical Research Letters*, 8, 05 2001. 119

[d'A08] Alexandre d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008. 97

[Dan16] Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 105–117, 2016. 124

[DDS14] Anindya De, Ilias Diakonikolas, and Rocco A. Servedio. Learning from Satisfying Assignments. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 478–497. SIAM, 2014. 52

[DFH$^+$15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126, 2015. 188

[DGN14] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014. 97

[DGR$^+$14] Nemanja Djuric, Mihajlo Grbovic, Vladan Radosavljevic, Narayan

Bhamidipati, and Slobodan Vucetic. Non-linear label ranking for large-scale prediction of long-term user interests. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. 39, 125

[DGT19a] Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. *Distribution-Independent PAC Learning of Halfspaces with Massart Noise*. Curran Associates Inc., Red Hook, NY, USA, 2019. 27

[DGT19b] Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. Distribution-independent pac learning of halfspaces with massart noise. *Advances in Neural Information Processing Systems*, 32, 2019. 27, 41, 124

[DGTZ18] Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Efficient Statistics, in High Dimensions, from Truncated Samples. In *59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 639–649. IEEE, 2018. 18, 26, 29, 31, 32, 33, 34, 37, 38, 51, 65, 68, 69, 83, 86, 98

[DGTZ19a] Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Computationally and Statistically Efficient Truncated Regression. In *Conference on Learning Theory (COLT)*, pages 955–960, 2019. 26, 31, 32, 51

[DGTZ19b] Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Computationally and statistically efficient truncated regression. In *Conference on Learning Theory*, pages 955–960. PMLR, 2019. 37, 86, 98

[DHK08] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory*, pages 355–366, 2008. 160

[DK19] Ilias Diakonikolas and Daniel M Kane. Recent advances in algorithmic high-dimensional robust statistics. *arXiv preprint arXiv:1911.05911*, 2019. 27

[DK20] Ilias Diakonikolas and Daniel M Kane. Hardness of learning halfspaces with massart noise. *arXiv preprint arXiv:2012.09720*, 2020. 41, 124

[DKFF13] Jia Deng, Jonathan Krause, and Li Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR

'13, page 580–587, USA, 2013. IEEE Computer Society. 34

[DKK+16] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust Estimators in High Dimensions without the Computational Intractability. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 655–664, 2016. 51, 54, 86, 120

[DKK+17] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being Robust (in High Dimensions) Can Be Practical. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 999–1008, 2017. 51, 87

[DKK+18] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustly Learning a Gaussian: Getting Optimal Error, Efficiently. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2683–2702, 2018. 51, 87

[DKK+19a] I. Diakonikolas, G. Kamath, D. Kane, J. Li, J. Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, pages 1596–1606, 2019. 87

[DKK+19b] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019. 27

[DKK+21] Ilias Diakonikolas, Daniel M Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning general halfspaces with general massart noise under the gaussian distribution. *arXiv preprint arXiv:2108.08767*, 2021. 124

[DKM05] Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *International conference on computational learning theory*, pages 249–263. Springer, 2005. 140

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 486–503. Springer, 2006. 47, 186

[DKPZ21] Ilias Diakonikolas, Daniel M Kane, Thanasis Pittas, and Nikos Zarifis. The optimality of polynomial regression for agnostic learning under gaussian marginals. *arXiv preprint arXiv:2102.04401*, 2021. 124

[DKS17a] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical Query Lower Bounds for Robust Estimation of High-dimensional Gaussians and Gaussian Mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84. IEEE, 2017. 33

[DKS17b] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84. IEEE, 2017. 87

[DKT21] Ilias Diakonikolas, Daniel Kane, and Christos Tzamos. Forster decomposition and learning halfspaces with noise. *Advances in Neural Information Processing Systems*, 34, 2021. 27, 124

[DKTZ20] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with massart noise under structured distributions. In *Conference on Learning Theory*, pages 1486–1513. PMLR, 2020. 27, 41, 122, 124

[DKTZ21] Constantinos Daskalakis, Vasilis Kontonis, Christos Tzamos, and Emmanouil Zampetakis. A statistical taylor theorem and extrapolation of truncated densities. In *Conference on Learning Theory*, pages 1395–1398. PMLR, 2021. 86

[DKZ20] Ilias Diakonikolas, Daniel Kane, and Nikos Zarifis. Near-optimal sq lower bounds for agnostically learning halfspaces and relus under gaussian marginals. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13586–13596. Curran Associates, Inc., 2020. 87, 124

[DOS18] Anindya De, Ryan O'Donnell, and Rocco Servedio. Learning sparse mixtures of rankings from noisy information. *arXiv preprint* *arXiv:1811.01216*, 2018. 40

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. 28, 47, 188

[DRV10] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010. 188, 198, 216

[DRZ20] Constantinos Daskalakis, Dhruv Rohatgi, and Manolis Zampetakis. Truncated linear regression in high dimensions. *arXiv preprint arXiv:2007.14539*, 2020. 37, 86, 98

[DSBDSS11] Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 207–232. JMLR Workshop and Conference Proceedings, 2011. 42, 155

[DSM03] Ofer Dekel, Yoram Singer, and Christopher D Manning. Log-linear models for label ranking. *Advances in neural information processing systems*, 16:497–504, 2003. 39, 124

[DSS14] Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Conference on Learning Theory*, pages 287–316. PMLR, 2014. 155

[dSSKC17] Cláudio Rebelo de Sá, Carlos Soares, Arno Knobbe, and Paulo Cortez. Label ranking forests. *Expert systems*, 34(1):e12166, 2017. 39, 124

[DV08] John Dunagan and Santosh Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. *Mathematical Programming*, 114(1):101–114, 2008. 36, 87

[Dwo08] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008. 188

[EDMMM06] Eyal Even-Dar, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine* *learning research*, 7(6), 2006. 162

[EKK$^+$22] Hossein Esfandiari, Alkis Kalavasis, Amin Karbasi, Andreas Krause, Vahab Mirrokni, and Grigoris Velegkas. Reproducible bandits. *arXiv preprint arXiv:2210.01898*, 2022. 46, 187

[EKM$^+$23] Hossein Esfandiari, Amin Karbasi, Vahab Mirrokni, Grigoris Velegkas, and Felix Zhou. Replicable clustering. *arXiv preprint arXiv:2302.10359*, 2023. 22, 46, 50

[EKMM21] Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab Mirrokni. Regret bounds for batched bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7340–7348, 2021. 162, 163, 164, 168, 170

[Fed13] Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 2013. 183

[Fel57] William Feller. An introduction to probability theory and its applications. *John Wiley*, 1957. 103

[Fel17] Vitaly Feldman. A general characterization of the statistical query complexity. In *Conference on Learning Theory*, pages 785–830. PMLR, 2017. 87

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 563–574. IEEE, 2006. 124

[FGR$^+$17] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM (JACM)*, 64(2):1–37, 2017. 36, 87

[FGRW12] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM J. Comput.*, 41(6):1558–1590, 2012. 27

[FGV17] Vitaly Feldman, Cristóbal Guzmán, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1265–1277, 2017. 36, 37, 87, 97

[FHMB08] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008. 39, 124

[FHT01] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001. 36, 97

[Fis31] RA Fisher. Properties and applications of Hh functions. *Mathematical tables*, 1:815–852, 1931. 31

[FJO$^+$15] Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Faster Algorithms for Testing under Conditional Sampling. In *Proceedings of the 28th Annual Conference on Learning Theory*, COLT '15, pages 607–636, 2015. 52

[FKKT21] Dimitris Fotakis, Alkis Kalavasis, Vasilis Kontonis, and Christos Tzamos. Efficient algorithms for learning from coarse labels. In *Conference on Learning Theory*, pages 2060–2079. PMLR, 2021. 22, 50, 209

[FKKT22] Dimitris Fotakis, Alkis Kalavasis, Vasilis Kontonis, and Christos Tzamos. Linear label ranking with bounded noise. *Advances in Neural Information Processing Systems*, 35:15642–15656, 2022. 22, 50

[FKP21] Dimitris Fotakis, Alkis Kalavasis, and Eleni Psaroudaki. Label ranking through nonparametric regression. *arXiv preprint arXiv:2111.02749*, 2021. 23, 50, 124, 125

[FKS21] Dimitris Fotakis, Alkis Kalavasis, and Konstantinos Stavropoulos. Aggregating incomplete and noisy rankings. In *International Conference on Artificial Intelligence and Statistics*, pages 2278–2286. PMLR, 2021. 23, 40, 50

[FKT20] Dimitris Fotakis, Alkis Kalavasis, and Christos Tzamos. Efficient parameter estimation of truncated boolean product distributions. In *Conference on Learning Theory*, pages 1586–1600. PMLR, 2020. 22, 50, 86

[FKT22] Dimitris Fotakis, Alkis Kalavasis, and Christos Tzamos. Perfect sampling from pairwise comparisons. *arXiv preprint arXiv:2211.12868*, 2022. 23, 50

[FLH$^+$20] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. *Advances in Neural Information Processing Systems*, 33:10948–10960, 2020. 26, 88

[FO85] Robert M Freund and James B Orlin. On the complexity of four polyhedral set containment problems. *Mathematical programming*, 33(2):139–145, 1985. 184

[FPV15] V. Feldman, W. Perkins, and S. Vempala. On the complexity of random satisfiability problems with planted solutions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC, 2015*, pages 77–86, 2015. 87

[FV86] Michael A Fligner and Joseph S Verducci. Distance Based Ranking Models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 359–369, 1986. 60

[Gal97] Francis Galton. An examination into the registered speeds of American trotting horses, with remarks on their value as hereditary data. *Proceedings of the Royal Society of London*, 62(379-387):310–315, 1897. 31

[GDGV13] Mihajlo Grbovic, Nemanja Djuric, Shengbo Guo, and Slobodan Vucetic. Supervised clustering of label ranking data using label preference information. *Machine learning*, 93(2-3):191–225, 2013. 124

[GDV12] Mihajlo Grbovic, Nemanja Djuric, and Slobodan Vucetic. Learning from pairwise preference data using gaussian mixture model. *Preference Learning: Problems and Applications in AI*, 33, 2012. 39, 124

[GFI16] Steven N Goodman, Daniele Fanelli, and John PA Ioannidis. What does research reproducibility mean? *Science translational medicine*, 8(341):341ps12–341ps12, 2016. 43

[GGJ$^+$20] Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In *International Conference on Machine Learning*, pages 3587–3596. PMLR, 2020. 87

[GGK20] Surbhi Goel, Aravind Gollakota, and Adam Klivans. Statistical-query lower bounds via functional gradients. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Ad-*

vances in Neural Information Processing Systems, volume 33, pages 2147–2158. Curran Associates, Inc., 2020. 87, 124, 209

[GGKM21] Badih Ghazi, Noah Golowich, Ravi Kumar, and Pasin Manurangsi. Sample-efficient proper pac learning with approximate differential privacy. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 183–196, 2021. 47, 188, 190, 196, 220, 221

[GHRU11] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 803–812, 2011. 209

[GHRZ19] Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32, 2019. 162, 163

[GKM21] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level differentially private learning via correlated sampling. *Advances in Neural Information Processing Systems*, 34:20172–20184, 2021. 28, 47, 188, 189, 195, 196, 197, 199, 206, 207, 208, 210, 217, 220, 221, 222, 223, 224

[GLB$^+$18] Yanming Guo, Yu Liu, Erwin M Bakker, Yuanhao Guo, and Michael S Lew. Cnn-rnn: a large-scale hierarchical image classification framework. *Multimedia tools and applications*, 77(8):10251–10271, 2018. 26, 34

[Gou00] Christian Gourieroux. *Econometrics of qualitative dependent variables*. Cambridge university press, 2000. 37

[GR09] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009. 124

[GTZ17] Themistoklis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Faster Sublinear Algorithms Using Conditional Sampling. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 1743–1757. SIAM, 2017. 52

[Gui09] Alice Guionnet. *Large random matrices*, volume 1957. Springer Science & Business Media, 2009. 112

[GVDLR97] Richard D Gill, Mark J Van Der Laan, and James M Robins. Coarsening at random: Characterizations, conjectures, counter-examples. In *Proceedings of the First Seattle Symposium in Biostatistics*, pages 255–294. Springer, 1997. 90

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001. 38, 98, 99, 100

[Hau18] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. In *The Mathematics of Generalization*, pages 37–116. CRC Press, 2018. 124

[HB06] Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006. 87

[HC15] Eyke Hüllermeier and Weiwei Cheng. Superset learning based on generalized loss minimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 260–275. Springer, 2015. 87

[HFCB08] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16):1897–1916, 2008. 39, 124, 125

[HIB$^+$18] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 43

[HKMV22] Steve Hanneke, Amin Karbasi, Shay Moran, and Grigoris Velegkas. Universal rates for interactive learning. *Advances in Neural Information Processing Systems*, 35:28657–28669, 2022. 190

[HL19] Samuel B Hopkins and Jerry Li. How Hard is Robust Mean Estimation? In *Conference on Learning Theory*, pages 1649–1682, 2019. 51, 87

[Hol07] Thomas Holenstein. Parallel repetition: simplifications and the no-signaling case. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 411–419, 2007. 206

[HPRZ03] Sariel Har-Peled, Dan Roth, and Dav Zimak. Constraint classification for multiclass classification and ranking. *Advances in neural information processing systems*, pages 809–816,

2003. URL: https://proceedings.neurips.cc/paper/2002/file/16026d60ff9b54410b3435b403afd226-Paper.pdf. 40, 121, 124

[HSSVG22] Daniel Hsu, Clayton Sanford, Rocco Servedio, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. Near-optimal statistical query lower bounds for agnostically learning intersections of halfspaces with gaussian marginals. *arXiv preprint arXiv:2202.05096*, 2022. 124

[Hub64] Peter J Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35:73–101, 1964. 27

[Hub99] Mark Huber. Efficient Exact Sampling from the Ising Model Using Swendsen-Wang. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '99, page 921–922, USA, 1999. Society for Industrial and Applied Mathematics. 69

[Hub04] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004. 86

[Hun04] David R Hunter. Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32(1):384–406, 2004. 40

[ILPS22] Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, page 818–831, New York, NY, USA, 2022. Association for Computing Machinery. 28, 43, 46, 47, 50, 161, 162, 163, 164, 181, 185, 186, 187, 188, 195, 199, 200, 205, 206, 208, 209, 210, 212, 214, 231, 232, 233, 234

[INHS17] Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary labels. *Advances in neural information processing systems*, 30, 2017. 87

[Ioa05] John PA Ioannidis. Why most published research findings are false. *PLoS medicine*, 2(8):e124, 2005. 43

[IZD20] Andrew Ilyas, Emmanouil Zampetakis, and Constantinos Daskalakis. A theoretical and practical framework for regression and classification from truncated samples. In *International Conference on Artificial Intelligence and Statistics*, pages 4463–4473. PMLR, 2020. 37, 86, 98

[JG02] Rong Jin and Zoubin Ghahramani. Learning with multiple labels.

Advances in neural information processing systems, 15, 2002. 87

[JKT20] Young Jung, Baekjin Kim, and Ambuj Tewari. On the equivalence between online and private learnability beyond binary classification. *Advances in neural information processing systems*, 33:16701–16710, 2020. 197, 198

[JLL+20] Qihan Jiao, Zhi Liu, Gongyang Li, Linwei Ye, and Yang Wang. Fine-grained image classification with coarse and fine labels on one-shot learning. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2020. 34

[JLYW19] Qihan Jiao, Zhi Liu, Linwei Ye, and Yang Wang. Weakly labeled fine-grained classification with hierarchy relationship of fine and coarse labels. *Journal of Visual Communication and Image Representation*, 63:102584, 2019. 26, 34

[KCG12] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *Artificial intelligence and statistics*, pages 592–600. PMLR, 2012. 160

[KCS17] Anna Korba, Stephan Clémençon, and Eric Sibony. A learning theory of ranking aggregation. In *Artificial Intelligence and Statistics*, pages 1001–1010. PMLR, 2017. 124

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998. 36, 41, 87, 88, 124, 209

[KGB18] Anna Korba, Alexandre Garcia, and Florence d'Alché Buc. A structured prediction approach for label ranking. *arXiv preprint arXiv:1807.02374*, 2018. 124

[KKM12] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory*, pages 199–213. Springer, 2012. 160

[KKM18] A. R. Klivans, P. K. Kothari, and R. Meka. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory, COLT 2018*, pages 1420–1430, 2018. 87

[KKMN09] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately.

In *Proceedings of the 18th international conference on World wide web*, pages 171–180, 2009. 215, 216

[KKMV23] Alkis Kalavasis, Amin Karbasi, Shay Moran, and Grigoris Velegkas. Statistical indistinguishability of learning algorithms. *arXiv preprint arXiv:2305.14311*, 2023. 22, 50

[KLN$^+$11] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011. 216

[KLSU18] Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. *arXiv preprint arXiv:1805.00216*, 2018. 59

[KMS06] Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors–a ptas for weighted feedback arc set on tournaments. In *ELECTRONIC COLLOQUIUM ON COMPUTATIONAL COMPLEXITY, REPORT NO. 144 (2006)*. Citeseer, 2006. 134

[KSS94] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2):115–141, 1994. 124

[KSZ22] Alkis Kalavasis, Konstantinos Stavropoulos, and Emmanouil Zampetakis. Learning and covering sums of independent random variables with unbounded support. *Advances in Neural Information Processing Systems*, 35:25185–25197, 2022. 23, 50

[KT02] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002. 206

[KT19] Gautam Kamath and Christos Tzamos. Anaconda: A Non-Adaptive Conditional Sampling Algorithm for Distribution Testing. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 679–693. SIAM, 2019. 52

[KTZ19] Vasilis Kontonis, Christos Tzamos, and Manolis Zampetakis. Efficient Truncated Statistics with Unknown Truncation. In *260th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1578–1595. IEEE, 2019. 26, 31, 32, 37, 51, 86, 98

[KVK22] Alkis Kalavasis, Grigoris Velegkas, and Amin Karbasi. Multiclass learnability beyond the pac framework: Universal rates and partial concept classes. *arXiv preprint arXiv:2210.02297*, 2022. 23, 50, 190

[KY05] Piyush Kumar and E Alper Yildirim. Minimum-volume enclosing ellipsoids and core sets. *Journal of Optimization Theory and applications*, 126(1):1–21, 2005. 183

[LB11] Tyler Lu and Craig Boutilier. Learning mallows models with pairwise preferences. In *ICML*, 2011. 40

[LBMK20] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020. 87

[LD14] Liping Liu and Thomas Dietterich. Learnability of the superset label learning problem. In *International Conference on Machine Learning*, pages 1629–1637. PMLR, 2014. 88

[Lee14] Alice Lee. Table of the Gaussian "Tail" Functions; When the "Tail" is Larger than the Body. *Biometrika*, 10(2/3):208–214, 1914. 31

[LGW17] Jie Lei, Zhenyu Guo, and Yang Wang. Weakly supervised image classification with coarse and fine labels. In *2017 14th Conference on Computer and Robot Vision (CRV)*, pages 240–247. IEEE, 2017. 26, 34

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988. 190, 201

[LKM$^+$18] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. *Advances in neural information processing systems*, 31, 2018. 43

[LM18] Allen Liu and Ankur Moitra. Efficiently learning mixtures of mallows models. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 627–638. IEEE, 2018. 40

[LM20] Roi Livni and Shay Moran. A limitation of the pac-bayes framework. *Advances in Neural Information Processing Systems*, 33:20543–20553, 2020. 188, 190

[LM21] Allen Liu and Ankur Moitra. Robust voting rules from algorithmic robust statistics. *arXiv preprint arXiv:2112.06380*, 2021. 40

[LP17a] Günter Last and Mathew Penrose. *Lectures on the Poisson process*, volume 7. Cambridge University Press, 2017. 193

[LP17b] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017. 206

[LRV16a] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016. 27

[LRV16b] Kevin A. Lai, Anup B. Rao, and Santosh Vempala. Agnostic Estimation of Mean and Covariance. In *IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674, 2016. 51, 87

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020. 160, 166, 169, 183

[LSW20] Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, pages 5662–5670. PMLR, 2020. 183

[Luc59] R.D. Luce. *Individual Choice Behavior*. Wiley, 1959. 60

[Luc12] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012. 40, 158

[LV07] László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007. 139

[LXF+20] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, pages 6500–6510. PMLR, 2020. 26, 88

[Mad86] Gangadharrao S Maddala. *Limited-dependent and qualitative variables in econometrics*. Number 3. Cambridge university press, 1986. 37

[Mal57] Colin L Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957. 40, 60, 158

[Mas07] Pascal Massart. *Concentration inequalities and model selection*, volume 6. Springer, 2007. 93

[McN14] Marcia McNutt. Reproducibility. *Science*, 343(6168):229–229, 2014. 43

[MKFI22] Jason Milionis, Alkis Kalavasis, Dimitris Fotakis, and Stratis Ioannidis. Differentially private regression with unbounded covariates. In *International Conference on Artificial Intelligence and Statistics*, pages 3242–3273. PMLR, 2022. 23, 50

[MM22] Maryanthe Malliaris and Shay Moran. The unstable formula theorem revisited. *arXiv preprint arXiv:2212.05050*, 2022. 190

[MN06] Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006. 27, 42, 124

[MS86] Olvi L Mangasarian and T-H Shiau. A variable-complexity norm maximization problem. *SIAM Journal on Algebraic Discrete Methods*, 7(3):455–461, 1986. 184

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007. 216

[MV19] Oren Mangoubi and Nisheeth K Vishnoi. Nonconvex sampling with the metropolis-adjusted langevin algorithm. In *Conference on Learning Theory*, pages 2259–2293. PMLR, 2019. 27, 41, 122, 124

[MW20] Cheng Mao and Yihong Wu. Learning mixtures of permutations: Groups of pairwise comparisons and combinatorial method of moments. *arXiv preprint arXiv:2009.06784*, 2020. 40

[Nat89] Balas K Natarajan. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989. 42, 155

[Naz03] Fedor Nazarov. On the maximal perimeter of a convex set in $\mathbb{R}^n$ with respect to a gaussian measure. In *Geometric aspects of functional analysis*, pages 169–187. Springer, 2003. 130, 147

[NC08a] Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–559, 2008. 87

[NC08b] Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–559, 2008. 87

[NDRT13] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013. 88

[NOS17] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287, 2017. 40

[NP19] Sai Ganesh Nagarajan and Ioannis Panageas. On the Analysis of EM for truncated mixtures of two Gaussians. In *31st International Conference on Algorithmic Learning Theory (ALT)*, pages 955–960, 2019. 26, 31, 37, 51, 86, 98

[NT22] Rajai Nasser and Stefan Tiegel. Optimal sq lower bounds for learning halfspaces with massart noise. *arXiv preprint arXiv:2201.09818*, 2022. 41, 124

[Owe88] Art B Owen. Empirical likelihood ratio confidence intervals for a single functional. *Biometrika*, 75(2):237–249, 1988. 89, 90

[Owe90] Art Owen. Empirical likelihood ratio confidence regions. *The Annals of Statistics*, 18(1):90–120, 1990. 89

[Owe01] Art B Owen. *Empirical likelihood*. CRC press, 2001. 89, 90

[Pap81] Christos H Papadimitriou. On the complexity of integer programming. *Journal of the ACM (JACM)*, 28(4):765–768, 1981. 141

[PCMY15] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015. 87

[PL08] Karl Pearson and Alice Lee. On the Generalised Probable Error in Multiple Normal Correlation. *Biometrika*, 6(1):59–68, 1908. 31

[Pla75] R. Plackett. The Analysis of Permutations. *Applied Statistics*, 24:193–202, 1975. 60

[PRMN04] Tomaso Poggio, Ryan Rifkin, Sayan Mukherjee, and Partha Niyogi.

General conditions for predictivity in learning theory. *Nature*, 428(6981):419–422, 2004. 188

[PSF+19] Joelle Pineau, Koustuv Sinha, Genevieve Fried, Rosemary Nan Ke, and Hugo Larochelle. Iclr reproducibility challenge 2019. *ReScience C*, 5(2), May 2019. 28, 43

[PVLS+21] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research: a report from the neurips 2019 reproducibility program. *Journal of Machine Learning Research*, 22, 2021. 28, 43

[QCJ+20] Zengyi Qin, Jiansheng Chen, Zhenyu Jiang, Xumin Yu, Chunhua Hu, Yu Ma, Suhua Miao, and Rongsong Zhou. Learning fine-grained estimation of physiological states from coarse-grained labels by distribution restoration. *Scientific Reports*, 10(1):1–10, 2020. 26, 34

[RBK07] Deva Ramanan, Simon Baker, and Sham Kakade. Leveraging archival video for building face datasets. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 87

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 34

[RdSRSK15] Cláudio Rebelo de Sá, Carla Rebelo, Carlos Soares, and Arno Knobbe. Distance-based decision tree algorithms for label ranking. In *Portuguese Conference on Artificial Intelligence*, pages 525–534. Springer, 2015. 39, 124

[RGGV15] M. Ristin, J. Gall, M. Guillaumin, and L. Van Gool. From categories to subcategories: Large-scale image classification with partial class label refinement. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 231–239, 2015. 34

[RRT+16] Maxim Raginsky, Alexander Rakhlin, Matthew Tsao, Yihong Wu, and Aolin Xu. Information-theoretic analysis of stability and bias of learning algorithms. In *2016 IEEE Information Theory Workshop (ITW)*, pages 26–30. IEEE, 2016. 189

[RS94] Ronald L Rivest and Robert Sloan. A formal model of hierarchical concept-learning. *Information and Computation*, 114(1):88–114, 1994. 27, 124

[SBG21] Satchit Sivakumar, Mark Bun, and Marco Gaboardi. Multiclass versus binary differentially private pac learning. *Advances in Neural Information Processing Systems*, 34:22943–22954, 2021. 197

[SBH13] Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference on learning theory*, pages 489–511. PMLR, 2013. 87

[Sch86] Helmut Schneider. *Truncated and censored samples from normal populations*. Marcel Dekker, Inc., 1986. 37

[Sch98] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998. 141

[Ser03] Rocco A Servedio. Smooth boosting and learning with malicious noise. *The Journal of Machine Learning Research*, 4:633–648, 2003. 231, 232, 233

[SFG$^+$09] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009. 46

[Sha18] Ohad Shamir. Distribution-specific hardness of learning neural networks. *The Journal of Machine Learning Research*, 19(1):1135–1163, 2018. 87

[SL22] Gil I Shamir and Dong Lin. Real world large scale recommendation systems reproducibility and smooth activations. *arXiv preprint arXiv:2202.06499*, 2022. 43

[Sli19] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019. 160

[Slo88] Robert Sloan. Types of noise in data for concept learning. In *Proceedings of the first annual Workshop on Computational Learning Theory*, pages 91–96, 1988. 27, 124

[Slo92] Robert H Sloan. Corrigendum to types of noise in data for concept learning. In *Proceedings of the fifth annual workshop on Computa-*

tional learning theory, page 450, 1992. 27, 124

[Slo96] Robert H Sloan. Pac learning, noise, and geometry. In *Learning and Geometry: Computational Approaches*, pages 21–41. Springer, 1996. 27, 124

[SS07] Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications.* PhD thesis, The Hebrew University of Jerusalem, 2007. 39, 124

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge university press, 2014. 41, 83, 117, 122, 201

[SV08] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008. 88

[SV16] Igal Sason and Sergio Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006, 2016. 201

[TG75] David R Thomas and Gary L Grunkemeier. Confidence interval estimation of survival probabilities for censored data. *Journal of the American Statistical Association*, 70(352):865–871, 1975. 90

[TGH15] Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42(2):245–284, 2015. 87

[TKD$^+$19] Fariborz Taherkhani, Hadi Kazemi, Ali Dabouei, Jeremy Dawson, and Nasser M Nasrabadi. A weakly supervised fine label classifier enhanced by coarse supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6459–6468, 2019. 34

[Tob58] James Tobin. Estimation of relationships for limited dependent variables. *Econometrica: journal of the Econometric Society*, pages 24–36, 1958. 37

[Tod16] Michael J Todd. *Minimum-volume ellipsoids: Theory and algorithms.* SIAM, 2016. 183

[TSD$^+$20] Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. Grafit: Learning fine-grained image representations with coarse labels. *arXiv preprint arXiv:2011.12982*, 2020. 26, 34

[Tsi06] Anastasios A. Tsiatis. *Semiparametric theory and missing data.* Springer Series in Statistics. Springer, New York, 2006. 26

[Tuk60] John Wilder Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485, 1960. 27

[UMR$^+$21] Enayat Ullah, Tung Mai, Anup Rao, Ryan A Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Conference on Learning Theory*, pages 4126–4142. PMLR, 2021. 189

[Vad17] Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017. 188

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 201

[Vap06] Vladimir Vapnik. *Estimation of dependences based on empirical data.* Springer Science & Business Media, 2006. 27, 124

[Vav93] Stephen A Vavasis. Polynomial time weak approximation algorithms for quadratic programming. In *Complexity in numerical optimization*, pages 490–500. World Scientific, 1993. 184

[VC15] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015. 202

[VEH20] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 87

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018. 113, 168

[VG10] Shankar Vembu and Thomas Gärtner. Label ranking algorithms: A survey. In *Preference learning*, pages 45–64. Springer, 2010. 124

[Vis21] Nisheeth K Vishnoi. *Algorithms for convex optimization.* Cambridge University Press, 2021. 138

[VRW17] Brendan Van Rooyen and Robert C Williamson. A theory of learning with corrupted labels. *J. Mach. Learn. Res.*, 18(1):8501–8550, 2017. 87, 88

[VW97] AW van der Vaart and Jon A Wellner. Weak convergence and empirical processes with applications to statistics. *Journal of the Royal Statistical Society-Series A Statistics in Society*, 160(3):596–608, 1997. 210

[VW19] Santosh Vempala and John Wilmes. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. In *Conference on Learning Theory*, pages 3115–3117. PMLR, 2019. 87

[VZW09] Anke Van Zuylen and David P Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Mathematics of Operations Research*, 34(3):594–620, 2009. 127, 134

[WCH$^+$21] Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In *International Conference on Machine Learning*, pages 11091–11100. PMLR, 2021. 26, 87, 88

[WDS19] Shanshan Wu, Alexandros G. Dimakis, and Sujay Sanghavi. Learning distributions generated by one-layer relu networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8105–8115, 2019. 37, 39

[Wol79] MS Wolynetz. Algorithm as 139: Maximum likelihood estimation in a linear model from confined and censored normal data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(2):195–206, 1979. 37, 86

[WZL21] Deng-Bao Wang, Min-Ling Zhang, and Li Li. Adaptive graph guided disambiguation for partial label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 87

[XLG19] Ning Xu, Jiaqi Lv, and Xin Geng. Partial label learning via label enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5557–5564, 2019. 87

[XQGZ21] Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. *Advances in Neural Information Processing Systems*, 34, 2021. 87

[Ye92] Yinyu Ye. On affine scaling algorithms for nonconvex quadratic programming. *Mathematical Programming*, 56(1):285–300, 1992. 184

[YZ16a] Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. In *Asian conference on machine learning*, pages 96–111. PMLR, 2016. 87

[YZ16b] Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. In *Asian conference on machine learning*, pages 96–111. PMLR, 2016. 87

[YZ17] Songbai Yan and Chicheng Zhang. Revisiting perceptron: Efficient and label-optimal learning of halfspaces. *Advances in Neural Information Processing Systems*, 30, 2017. 27, 124

[Zha14] Min-Ling Zhang. Disambiguation-free partial label learning. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 37–45. SIAM, 2014. 87

[ZL21] Chicheng Zhang and Yinan Li. Improved algorithms for efficient active learning halfspaces with massart and tsybakov noise. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 4526–4527. PMLR, 15–19 Aug 2021. 27, 41, 122, 124

[ZLC17] Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022. PMLR, 2017. 27, 124

[ZLGQ14] Yangming Zhou, Yangguang Liu, Xiao-Zhi Gao, and Guoping Qiu. A label ranking method based on gaussian mixture model. *Knowledge-Based Systems*, 72:108–113, 2014. 39, 124

[ZLY$^+$14] Yangming Zhou, Yangguang Liu, Jiangang Yang, Xiaoqi He, and Liangliang Liu. A taxonomy of label ranking algorithms. *J. Comput.*, 9(3):557–565, 2014. 124

[ZQ18] Yangming Zhou and Guoping Qiu. Random forest for label ranking. *Expert Systems with Applications*, 112:99–109, 2018. 39, 124

[ZSA20] Chicheng Zhang, Jie Shen, and Pranjal Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7184–7197. Curran Associates, Inc., 2020. 27, 41, 122, 124, 126, 133

[ZY15] Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *Twenty-fourth international joint conference on artificial intelligence*, 2015. 87

[ZYT17a] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2155–2167, 2017. 87

[ZYT17b] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2155–2167, 2017. 87

[ZZL16] Min-Ling Zhang, Bin-Bin Zhou, and Xu-Ying Liu. Partial label learning via feature-aware disambiguation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1335–1344, 2016. 87