Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

Τεχνικές Εκτίμησης Ιδιωτικών Παραμέτρων σε Μη-Φιλαληθείς Δημοπρασίες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΡΗΓΟΡΗΣ ΒΕΛΕΓΚΑΣ

Επιβλέπων : Δημήτρης Φωτάκης
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2018

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

Τεχνικές Εκτίμησης Ιδιωτικών Παραμέτρων σε Μη-Φιλαληθείς Δημοπρασίες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΡΗΓΟΡΗΣ ΒΕΛΕΓΚΑΣ

Επιβλέπων : Δημήτρης Φωτάκης
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 17η Οκτωβρίου 2018.

Δημήτρης Φωτάκης Αν. Καθηγητής Ε.Μ.Π.	Αριστείδης Παγουρτζής Αν. Καθηγητής Ε.Μ.Π.	Νικόλαος Παπασπύρου Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2018

Γρηγόρης Βελέγκας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Οι ηλεκτρονικές δημοπρασίες διαφημίσεων, οι οποίες πουλάνε διαφημιστικό χώρο δίπλα από το αποτελέσματα των οργανικών αναζητήσεων, φέρνουν έσοδα δισεκατομμυρίων δολλαρίων. Στις δημοπρασίες για λέξεις κλειδιά όποτε ο χρήστης κάνει αναζήτηση για κάποια λέξη κλειδί, οι διαφημιστές κάνουν μια προσφορά στη μηχανή αναζήτησης και αφού η μηχανή αναζήτησης συλλέξει όλες τις προσφορές, αποφασίζει, λαμβάνοντας υπόψιν και κάποιους άλλους παράγοντες, πώς θα κατανείμει τον διαφημιστικό χώρο. Ένα κοινό χαρακτηριστικό των περίσσοτερων διαφορετικών τύπων δημοπρασιών για λέξεις κλειδιά που χρησιμοποιούνται στην πράξη είναι η μη-*φιλαληθεία*, κάτι που σημαίνει ότι η προσφορά που κάνουν οι διαφημιζόμενοι δεν είναι η *ωφελεία* τους, δηλαδή το πόσο πραγματικά θέλουν το κλικ του χρήστη. Ένα μείζον πρόβλημα για τον δημοπράτη σε τέτοια συστήματα είναι η εκτίμηση αυτής της παραμέτρου, της *ωφελείας*, χρησιμοποιώντας δεδομένα από προηγούμενες δημοπρασίες στις οποίες έχει συμμετάσχει ο διαφημιζόμενος. Οι κλασσικές προσεγγίσεις σε αυτό το πρόβλημα υποθέτουν ότι οι διαφημιζόμενοι έχουν φτάσει σε μια σταθερή κατάσταση όπου ο καθένας αντιδρά με βέλτιστο τρόπο στις στρατηγικές αποφάσεις των υπολοίπων. Δυστυχώς, αυτή η υπόθεση είναι λανθασμένη στις δυναμικές αγορές όπου επικεντρωνόμαστε.

Στην παρούσα διπλωματική παρουσίζουμε μια νέο μέθοδο για την επίλυση αυτού του προβλήματος που προτάθηκε από τους Nekipelov κ.α., η οποία βασίζεται στην πολύ ασθενέστερη υπόθεση ότι οι συμμετέχοντες στις δημοπρασίσες χρησιμοποιούν αλγοριθμικές τεχνικές μάθησης και πετυχαίνουν τον στόχο της *μη-μετάνοιας*. Στο μοντέλο που χρησιμοποιούν υποθέτουν ότι οι ωφελείες των διαφημιζομένων παραμένουν σταθερές στο πέρασμα του χρόνου. Παρουσίαζουμε μια επέκταση της μεθόδου τους στην οποία οι ωφελείες μεταβάλλονται σχετικά αργά κατά τη διάρκεια των δημοπρασιών. Επιπλέον, παρουσίαζουμε ένα τρόπο χρήσης της μεθόδου τους σε περιβάλλοντα δημοπρασιών ενός χρήστη-ενός αντικειμένου όπου η παράμετρος της ωφελείας δεν αλλάζει, ώστε ο δημοπράτης να θέσει κατάλληλα τις *τιμές κράτησης* για να βελτιστοποιήσει το κέρδος του.

Στο τρίτο κεφάλαιο εισάγωγουμε τον αναγνώστη στις βασικές έννοιες του Σχεδιασμού Μηχανισμών τις οποίες θα χρησιμοποίησουμε εκτεταμένα σε όλο το κείμενο. Στο τέταρτο κεφάλαιο εισάγουμε τις βασικές έννοιες και τους αλγορίθμους που χρησιμοποίουνται από τους παίχτες σε περιβάλλοντα online αποφάσεων, η οποία θα είναι και η βασική μας υπόθεση συμπεριφοράς για το πώς λειτουργούν οι αγοραστές σε μη-φιλαληθείς δημοπρασίες. Στο πέμπτο κεφάλαιο παρουσιάζουμε τη μέθοδο εκτίμησης των Nekipelov κ.α., καθώς και κάποια αποτελέσματα πρόσφατων δημοσιεύσεων που στοχεύουν να αναλύσουν αγορές κάτω από υποθέσεις μάθησης. Τέλος, στο έκτο κεφάλαιο παρουσιάζουμε τα αποτελέσματα της δικής μας προσπάθειας.

Λέξεις κλειδιά

αλγοριθμική θεωρία παιγνίων, σχεδίαση μηχανισμών, ηλεκτρονικές δημοπρασίες, ευφυείς πράκτορες, μηχανιστική μάθηση, εκτίμηση ωφέλειας

Abstract

Online ad auctions, which sell advertising space online alongside the organic results, generate billions of revenue each year. In the keyword auctions, whenever a user searches for a specific keyword the advertisers submit a bid to the search engine and, based on these bids along with some other factors, the engine decides how to allocate the advertising space. A common characteristic of the different auction formats that are used in practice is that they are not *truthful*, which means that it is not of the best interest of the advertisers to submit to the allocation mechanism their *valuation*, i.e. how much they value getting clicked. A major problem for the auctioneers in such settings, is to manage to infer that parameter based on the past data that they have collected from previous auctions. Classical work on this problem assumed that the advertisers have managed to somehow reach a stable state, in which each one best responds to the strategies of his opponents. However, this assumption is unrealistic in dynamic markets that we are interested in.

In this thesis we present the new approach to that problem that was suggested by Nekipelov et al. [NST15], which is based on the much weaker assumption that advertisers are learning agents who achieve the *no-regret* task. In their model, they assume that the bidders' valuations remain constant over time. We extend their results in settings in which these valuations are slowly changing throughout the repeated auctions. We also show how to use the valuation inferring method that was proposed by Nekipelov et al. to set the reserve prices in single item-single buyer settings in order to maximize his revenue , when the valuation of the buyer does not change.

In the third chapter we introduce the reader to some basic mechanism design concepts that we will use extensively in the following chapters. In the fourth chapter we introduce basic concepts and algorithms used by players in online decision settings which will constitute our basic assumption about how players behave in non-truthful auction settings. In the fifth chapter we present the inference method proposed by Nekipelov et al. as well as some other results from the recent line of work that aims to analyze repeated games under learning assumptions. Finally, in the sixth chapter we analyze our results regarding the extension of that valuation inference method.

Key words

algorithmic game theory, mechanism design, sponsored search auctions, GSP, online convex optimization, online learning, valuation inference

Ευχαριστίες

Η ολοκληρώση της διπλωματικής εργασίας σηματοδοτεί και το τέλος των προπτυχιακών μου σπουδών. Αισθάνομαι λοιπόν την ανάγκη να ευχαριστήσω τους ανθρώπους που ήταν δίπλα μου αυτά τα χρόνια.

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου, κ. Δημήτρη Φωτάκη. Αισθάνομαι τυχερός όχι μόνο για τη συνεργασία μας κατά τη διαρκεία αυτής της εργασίας, όπου παρότι δεν βρισκόταν στην Ελλάδα μπόρεσε να αφιερώσει σημαντικό χρόνο και έδειξε από την αρχή μεγάλη εμπιστοσύνη στο πρόσωπό μου, αλλά κυρίως για την αγάπη που μου καλλιέργησε για τη θεωρητική πληροφορική, από τα πρώτα κιόλας έτη των σπουδών μου, μέσα από τον δικό του ενθουσιασμό και τρόπο διδασκαλίας.

Πολύ σημαντικό ρόλο στην πορεία μου έχει παίξει η οικογένειά μου που με στηρίζει διαρκώς σε όλη μου τη ζωή, γι' αυτό και την ευχαριστώ θερμά. Τέλος, θέλω να εκφράσω τις ευχαριστίες μου στους φίλους μου και ιδιαίτερα στον Κυριάκο και στον Παναγιώτη, με τους οποίους περάσαμε ατελείωτες ώρες αυτά τα χρόνια. Είμαι σίγουρος ότι όλο αυτό το διάστημα θα ήταν λιγότερο ευχάριστο χωρίς την παρουσία τους.

Γρηγόρης Βελέγκας,

Αθήνα, 17η Οκτωβρίου 2018

Περιεχόμενα

Κατάλογος σχημάτων

Κεφάλαιο 1

Εκτεταμένη Ελληνική Περίληψη

Στο σημείο αυτό θα συνοψίσουμε το περιεχόμενο της παρούσας διπλωματικής, δίνοντας βασικούς ορισμούς και θεωρήματα, χωρίς αποδείξεις.

1.1 Σημεία Ισορροπίας και Σχεδιασμός Μηχανισμών

1.1.1 Σημεία Ισορροπίας

Η Θεωρία Παιγνίων μελετά την αλληλεπίδραση στρατηγικών οντοτήτων. Πολύ σημαντικό ρόλο στην αλληλεπίδραση αυτή παίζουν τα λεγόμενα "σημεία ισορροπίας". Διαισθήτικα, λέμε ότι ένα παίγνιο είναι σε κάποιο σημείο ισορροπίας όταν οι συμμετέχοντες σε αυτό δεν έχουν λόγο να διαφοροποιήσουν τη στρατηγική τους, οπότε καθώς ο χρόνος περνά, οι κινήσεις που κάνουν δεν αλλάζουν. Έτσι, υπάρχει κατά κάποιο τρόπο ισορροπία. Για να δούμε βασικά παραδείγματα τέτοιων ισορροπιών είναι χρήσιμο να ορίσουμε ένας είδος παιγνίου.

Ορισμός: Παίγνιο Ελαχιστοποίησης Κόστους

- n παίχτες
- ένας πεπερασμένος *στρατηγικός χώρος* S_i για κάθε παίχτη i. Στο πλαίσιο των δημοπρασιών, αυτός ο χώρος είναι οι διαφορετικές προσφορές που μπορεί να κάνει ο παίχτης. Συμβολίζουμε με $\boldsymbol{s} = (s_1, ..., s_n)$ το *στρατηγικό προφίλ* του παιγνίου
- μια *συνάρτηση κόστους* $C_i : S_1 \times ... \times S_n \to R$ για κάθε παίχτη i (ο καθένας θέλει να ελαχιστοποιήσει το κόστος του)

Παρακάτω παραθέτουμε μερικές από τις πιο σημαντικές έννοιες ισορροπίας.

- **Κυρίαρχη Στρατηγική**
  Η στρατηγική s' για τον παίχτη i είναι κυρίαρχη $\forall s \in S_i, \forall \boldsymbol{s}_{-i} \in S_1 \times ... \times S_{i-1} \times S_{i+1} \times ... \times S_n$ ισχύει
  $$C_i(s', \boldsymbol{s}_{-i}) \leq C_i(s, \boldsymbol{s}_{-i})$$
- **Ανάμεικτη Ισορροπία Nash**
  Για κάθε παίχτη και κάθε άλλη στρατηγική του ισχύει
  $$\mathbb{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})] \leq \mathbb{E}_{\boldsymbol{s_{-i}}\sim\sigma_{-i}}[C_i(s_i', \boldsymbol{s_{-i}})]$$
- **Καθαρή Ισορροπία Nash**
  Για κάθε παίχτη και κάθε αλλη στρατηγική ισχύει
  $$C_i(\boldsymbol{s}) \leq C_i(s_i', \boldsymbol{s_{-i}})$$

- **Συσχετισμένη Ισορροπία**
  Για κάθε παίχτη και κάθε αλλη στρατηγική ισχύει

$$\mathbb{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})|s_i] \leq \mathbb{E}_{\boldsymbol{s_{-i}}\sim\sigma_{-i}}[C_i(s_i', \boldsymbol{s_{-i}})|s_i]$$

- **Χαλαρά Συσχετισμένη Ισορροπία**
  Για κάθε παίχτη και κάθε αλλη στρατηγική ισχύει

$$\mathbb{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})] \leq \mathbb{E}_{\boldsymbol{s_{-i}}\sim\sigma_{-i}}[C_i(s_i', \boldsymbol{s_{-i}})]$$

1.1.2 Σχεδιασμός Μηχανισμών

Ο Σχεδιασμος Μηχανισμών ασχολείται με τη δημιουργία κανόνων και συστημάτων που διέπουν τη λειτουργία τέτοιων στρατηγικών οντοτήτων. Στόχος είναι ο σχεδιασμός κανόνων που δίνουν στους παίκτες εύκολα ανιχνεύσιμες *κυρίαρχες στρατηγικές*. Με αυτό τον τρόπο είναι εύκολο για τους παίχτες να αποφασίσουν πώς θα παίξουν, αλλά και για τον δημιουργό να προβλέψει το αποτέλεσμα της αλληλεπίδρασης. Ένα παράδειγμα τέτοιου μηχανισμού είναι η δημοπρασία Δεύτερης Τιμής.

Δημοπρασία Δεύτερης Τιμής
Έστω ότι θέλουμε να πουλήσουμε ένα αντικείμενο σε ένα σύνολο από n αγοραστές. Ζητάμε από όλους να μας δώσουν μια προσφορά ιδιωτικά και δίνουμε το αντικείμενο σε αυτόν που έκανε τη μεγαλύτερη προσφορά και κατόπιν τον χρεώνουμε τη δεύτερη μεγαλύτερη προσφορά. Ο λόγος γι' αυτή τη φαινομενικά περίεργη τακτική είναι ότι με αυτό τον τρόπο καταφέρνουμε να δώσουμε κίνητρο στους αγοραστές να μας πουν την αλήθεια για το πόσο πραγματικά είναι διατεθημένοι να πληρώσουν για το αντικείμενο. Μηχανισμοί στους οποίους το να λένε οι παίχτες την αλήθεια είναι κυρίαρχη στρατηγική λέγονται φιλαληθείς. Αυτή είναι μια πολύ σημαντική ιδιότητα των μηχανισμών.

Περιβάλλον Μονής Παραμέτρου
Χαρακτηρίζουμε περίβαλλοντα μονής παραμέτρου αυτά στα οποία η ικανοποίηση των συμμετεχόντων έχει τη μορφή $u(b) = v \cdot x(b) - c(b)$, όπου $x(), p()$ είναι η συνάρτηση τοποθέτησης και πληρωμής αντίστοιχα, τις οποίες επιλέγει ο μηχανισμός. Οι μηχανισμοί μονής παραμέτρου παίζουν πολύ σημαντικό ρόλο και έχουν μελετηθεί εκτενώς. Ένα πολύ σημαντικό αποτέλεσμα είναι το Λήμμα του Myerson που παρατίθεται παρακάτω.

Λήμμα Myerson Στο περιβάλλον μονής παραμέτρου ισχύουν τα ακόλουθα

1. Ένας κανόνας τοποθέτησης $\boldsymbol{x}$ μπορεί να γίνει φιλαληθής αν και μόνο αν είναι μονότονος.
2. Αν ένας κανόνας τοποθέτησης είναι μονότονος υπάρχει ένας κανόνας πληρωμής $\boldsymbol{p}$ ώστε ο μηχανισμός $(\boldsymbol{x}, \boldsymbol{p})$ να είναι φιλαληθής.
3. Ο τύπος για τον κανόνα πληρωμής είναι $p_i(b_i, \boldsymbol{b_{-i}}) = \int_0^{b_i} z \cdot \frac{d}{dz} x_i(z, \boldsymbol{b_{-i}}) dz$.

Γενικευμένες Δημοπρασίες Δεύτερης Τιμής Με τη χρήση των γενικευμένων δημοπρασιών δεύτερης τιμής πωλείται διαφημιστικός χώρος στις αναζητήσεις για λέξεις κλειδιά στο διαδίκτυο. Παραθέτουμε τα βασικά συστατικά τους.

1. n διαφημιζόμενοι
2. m διαφημιστικές θέσεις
3. v_i ιδιωτική παράμετρος για τον καθένα

4. $\boldsymbol{a} = (a_1, ..., a_m)$ οι συντελεστές θέσης $a_1 \geq a_2 \geq ... \geq a_m$ που υποθέτουμε ότι είναι σε φθίνουσα σειρά

5. $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_n)$ οι πιθανότητες κλικ των διαφημιζομένων

6. $\boldsymbol{s} = (s_1, ..., s_n)$, οι συντελεστές ποιότητας των διαφημιζομένων

7. r, το σκορ κράτησης της δημοπρασίας

8. $\boldsymbol{b} = (b_1, ..., b_n)$ οι προσφορές που έγιναν από τους παίχτες

Οι διαφημιζόμενοι ταξινομούνται σύμφωνα με το σκορ $s_i b_i$ και παίρνουν θέσεις σε φθίνουσα σειρά. Κάθε φορά που ο χρήστης πατάει τις διαφημίσεις τους χρεώνονται την ελάχιστη τιμή που θα πρεπε να εχουν προσφέρει για να διατηρήσουν τη θέση τους. Το πρόβλημα είναι ότι αυτός ο μηχανισμός δεν είναι φιλαληθής. Γι' αυτό το λόγο οι παίχτες θα πρέπει να εφαρμόσουν κάποιους αλγόριθμους μηχανικής μάθησης ώστε να μάθουν να παίζουν σωστά.

1.2 Online Μάθηση

Η online μάθηση είναι ένας τομέας της μηχανικής μάθησης που ασχολείται με τη λήψη ακολουθιακών αποφάσεων σε ενα άγνωστο περιβάλλον, χωρίς κάποιες υποθέσεις για πιθανοτική κατανομή. Το μοντέλο έχει την ακόλουθη μορφή

- Σε κάθε γύρο t διαλέγουμε ένα σημείο x_t
- Ο αντίπαλος διαλέγει μια συνάρτηση $f_t()$
- Χάνουμε $f_t(x_t)$

1.2.1 Στατικό no-regret

Στόχος του παίχτη που χρησιμοποιεί ένα τέτοιο αλγόριθμο είναι να ελαχιστοποιήσει το regret του που ορίζεται ως εξής.

Regret

$$\text{regret}_T(A) = \sup_{\{f_1,...,f_T\} \subseteq F} \{ \sum_{t=1}^{T} f_t(\boldsymbol{x_t}) - \min_{\boldsymbol{x} \in K} \sum_{t=1}^{T} f_t(\boldsymbol{x}) \}$$

Το regret μετράει το πόσο μετανιώνει ο παίχτης ότι δεν έπαιζε συνέχεια το καλύτερο σταθερό σημείο του συνόλου. Για να θεωρήσουμε ότι κάποιος έχει no-regret θα πρέπει η παραπάνω ποσότητα να είναι $o(T)$. Θεωρούμε ότι το σύνολο απόφασης είναι κυρτό και οι συναρτήσεις του αντιπάλου είναι επίσης κυρτές. Δίνουμε τον ορισμό παρακάτω.

Κυρτό Σύνολο

Ένα σύνολο $K \subseteq \mathbb{R}^n$ είναι κυρτό αν $\forall \boldsymbol{x}, \boldsymbol{y} \in K, \forall a \in [0,1]$ ισχύει ότι $a\boldsymbol{x} + (1-a)\boldsymbol{y} \in K$, δηλαδή τα σημεία που βρίσκονται στη γραμμή που ενώνει δύο σημεία του συνόλου βρίσκονται επίσης στο σύνολο.

Κυρτή Συνάρτηση

Μια συνάρτηση $f : K \to \mathbb{R}$ είναι κυρτή αν $\forall \boldsymbol{x}, \boldsymbol{y} \in K, \forall a \in [0,1]$ ισχύει ότι $f(a\boldsymbol{x} + (1-a)\boldsymbol{y}) \leq af(\boldsymbol{x}) + (1-a)f(\boldsymbol{y})$.

Υπάρχουν πολλοί αλγόριθμοι που επιτυγχάνουν no-regret. Όταν το σύνολο απόφαση είναι διακριτό πολύ χρήσιμος είναι ο παρακάτω.

Algorithm 1 Multiplicative Weights Update

Initialize: $\forall i \in [N], W_1(i) = 1$
for $t = 1...T$ **do** ▷ We have to answer T questions
    receive question q_t
    pick i_t according to W_t, i.e. $\mathbb{P}[i_t = i] = \boldsymbol{x}_t(i) = \frac{W_t(i)}{\sum_{j=1}^{n} W_t(j)}$
    suffer loss $l_t(i_t)$
    update weights $W_{t+1}(i) = W_t(i)e^{-\epsilon l_t(i)}, \forall i$

Η διαίσθηση πίσω από τον προηγούμενο αλγόριθμο είναι ότι όλη η μάζα πιθανότητας που μοιράζουμε μεταξύ των διαφορετικών σημείων του χώρου μαζεύεται γρήγορα γύρω από το σημείο που έχει καλή επίδοση.

Όταν ο χώρος απόφασης είναι συνεχής ένας σημαντικός αλγόριθμος είναι ο παρακάτω.

Algorithm 2 Online Gradient Descent

for $t = 1...T$ **do** ▷ We make T iterations
    play $\boldsymbol{x}_t$ and observe loss $f_t(\boldsymbol{x}_t)$
    $\boldsymbol{y}_t = \boldsymbol{x}_t - \eta_t \nabla f_t(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} = \Pi_K(\boldsymbol{y}_{t+1})$ ▷ Project to K to maintain feasibility
return $\boldsymbol{x}_{T+1}$

Η έμπνευση του έρχεται από τον τομέα της Κυρτής Βελτιστοποίησης στον οποίο κατέχει κεντρικό ρόλο.

1.2.2 Δυναμικό no-regret

Σε περίπτωση που οι συναρτήσεις τις οποίες πρέπει να "μάθει" ο παίχτης δεν αλλάζουν συχνά στο χρόνο, τότε μπορεί η διαδικασία μάθησης να έχει ισχυρότερες εγγυήσεις επίδοσης. Ορίζουμε πρώτα της παρακάτω μετρικές που δείχνουν πόσο μεγάλες είναι οι αλλαγές στο περιβάλλον.

$$V_T = \sum_{t=2}^{T} \sup_{\boldsymbol{x} \in K} |f_t(\boldsymbol{x}) - f_{t-1}(\boldsymbol{x})|$$

Η παραπάνω ποσότητα μετράει το πόσο αλλάζουν οι συναρτήσεις που μας δίνει ο αντίπαλος.

$$C_T(\boldsymbol{u}_1, ..., \boldsymbol{u}_T) = \sum_{t=2}^{T} ||\boldsymbol{u}_t - \boldsymbol{u}_{t-1}||$$

Αυτή η ποσότητα μετράει το πόσο αλλάζει η ακολουθία των σημείων που ανταγωνιζόμαστε. Το δυναμικό regret ορίζεται ως εξής.

$$\text{regret}_T^d(\boldsymbol{x}_1^*, ..., \boldsymbol{x}_T^*) = \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{x}_t^*)$$

Όλοι οι αλγόριθμοι που πετυχαίνουν $o(T)$ dynamic regret έχουν το κοινό χαρακτηριστικό ότι μεροληπτούν και δίνουν μεγαλύτερο βάρος στις πρόσφατες εισόδους των αλγορίθμων έναντι των παλαιών. Παρακάτω φαίνονται τα όρια του dynamic regret για διάφορες περιπτώσεις.

Regret notion	Loss function	Regret rate
$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t)$	Convex	$\mathcal{O}\left(\sqrt{T}(1 + C_T(\mathbf{u}_1, \ldots, \mathbf{u}_T))\right)$
$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t)$	Convex	$\mathcal{O}\left(\sqrt{T}(1 + C'_T(\mathbf{u}_1, \ldots, \mathbf{u}_T))\right)$
$\sum_{t=1}^{T} \mathbb{E}\left[f_t(\mathbf{x}_t)\right] - f_t(\mathbf{x}_t^*)$	Convex	$\mathcal{O}\left(T^{2/3}(1 + V_T)^{1/3}\right)$
$\sum_{t=1}^{T} \mathbb{E}\left[f_t(\mathbf{x}_t)\right] - f_t(\mathbf{x}_t^*)$	Strongly convex	$\mathcal{O}\left(\sqrt{T(1 + V_T)}\right)$
$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)$	Convex	$\mathcal{O}\left(\sqrt{D_T + 1} + \min\left\{\sqrt{(D_T + 1)C_T}, [(D_T + 1)V_T T]^{1/3}\right\}\right)$
$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)$	Strongly convex	$\mathcal{O}\left(1 + C_T\right)$

Σχήμα 1.1: Όρια δυναμικού regret

1.2.3 Εξαγωγή Παραμέτρων Ωφελείας σε Δημοπρασίες για Λέξεις Κλειδιά

Ένα πολύ σημαντικό πρόβλημα σε δημοπρασίες για λέξεις κλειδιά, αλλά και μη-φιλαληθείς μηχανισμούς γενικότερα, είναι η εκτίμηση της παραμέτρου v του διαφημιζόμενου. Οι κλασικές προσεγγίσεις για αυτό το πρόβλημα υπέθεταν ότι οι παίχτες έχουν φτάσει σε μια ισορροπία Nash, δηλαδή ότι ο καθένας αντιδρά με βέλτιστο τρόπο στις στρατηγικές αποφάσεις των αντιπάλων του. Κάτι τέτοιο ωστόσο δεν είναι ρεαλιστικό στις μεγάλες και δυναμικές αγορές που μελετάμε, αφού από τη μία η εύρεση μιας τέτοιας ισορροπίας είναι ένα υπολογιστικά δύσκολο πρόβλημα, και από την άλλη ο όγκος της πληροφορίας που χρειάζεται να ανταλλαχθεί μεταξύ των παικτών είναι πολύ μεγάλος. Μια μέθοδος που προτάθηκε πρόσφατα από τους Nekipelov κ.α. βασίζεται στην ασθενέστερη υπόθεση ότι οι παίχτες χρησιμοποιούν αλγόριθμους μάθησης όπως αυτούς που περιγράψαμε πριν. Αρχικά θα δώσουμε κάποιους ορισμούς.

- $$\Delta P(b') = \frac{1}{T}\sum_{t=1}^{T}(p_{it}(b', \boldsymbol{b}_{-i}) - p_{it}(\boldsymbol{b}))$$

  Αυτή η ποσότητα μετράει το πόσο θα μεταβληθεί η πιθανότητα του κλικ αν ο παίχτης άλλαζε την ακολουθία των πονταρισμάτων του και πόνταρε συνέχεια b'.

- $$\Delta C(b') = \frac{1}{T}\sum_{t=1}^{T}(c_{it}(b', \boldsymbol{b}_{-i}) - c_{it}(\boldsymbol{b}))$$

  Αυτή η ποσότητα μετράει το πόσο θα μεταβληθεί το κόστος του κλικ αν ο παίχτης άλλαζε την ακολουθία των πονταρισμάτων του και πόνταρε συνέχεια b'.

Χρησιμοποιώντας αυτούς τους ορισμούς λέμε ότι ένα ζευγάρι ωφελείας, regret είναι "λογικό" όταν ισχύει $v\Delta P(b) \leq \Delta C(b) + \epsilon, \forall b \in B$. Αυτή η συνθήκη μας λέει ότι αν ο παίχτης είχε παράμετρο ωφελείας v τότε το *regret* του θα ήταν το πολύ ϵ. Το σύνολο αυτών των σημείων ονομάζεται "λογικό" σύνολο. Ισχύουν τα παρακάτω δύο θεωρήματα σχετικά με αυτό το σύνολο.

Θεώρημα: Το σύνολο των ζευγαριών που ικανοποιούν αυτή την ανισότητα είναι κλειστό και κυρτό.

Θεώρημα: Για κάθε regret ϵ το σύνολο των παραμέτρων ωφελείας που είναι λογικές για αυτό το regret δίνεται από τη σχέση

$$v \in \left[\max_{b':\Delta P(b')<0} \frac{\Delta C(b') + \epsilon}{\Delta P(b')}, \min_{b':\Delta P(b')>0} \frac{\Delta C(b') + \epsilon}{\Delta P(b')}\right]$$

Το παραπάνω θεώρημα μας δίνει ένα τρόπο να εκτιμήσουμε το λογικό σύνολο από τα δεδομένα. Κάνουμε μια διακριτοποίηση στο χώρο των ϵ και b και έτσι μπορούμε να υπολογίσουμε όλες τις παραμέτρους που υπάρχουν στο παραπάνω θεώρημα.

1.3 Αποτελέσματα

Το βασικό πρόβλημα με το οποίο ασχοληθήκαμε είναι η εξαγωγή ιδιωτικών παραμέτρων σε περιβάλλοντα δημοπρασιών για λέξεις κλειδιά, οι οποίες δεν ικανοποιούν την ιδιότητα της φιλαληθείας. Στηριχθήκαμε στη μέθοδο που παρουσιάσαμε στην προηγούμενη ενότητα, όπου διατηρήσαμε την υπόθεση ότι οι παίχτες χρησιμοποιούν κάποια διαδικασία μάθησης ώστε να βελτιώνονται στο χρόνο. Η μέθοδος των Nekipelov κ.α. [NST15] υποθέτει ότι η παράμετρος της ωφελείας παραμένει σταθερή στο χρόνο. Κάτι τέτοιο όμως δεν είναι ρεαλιστικό σε μεγάλες αγορές, όπου η εποχικότητα και άλλοι παράγοντες παίζουν μεγάλο ρόλο. Για παράδειγμα, ένα ανθοπωλείο την ημέρα του Αγίου Βαλεντίνου έχει πολύ μεγαλύτερα κέρδη από τις επισκέψεις των χρηστών απ' ότι μια οποιαδήποτε άλλη μέρα του χρόνου. Για αυτό το λόγο προτείνουμε μια επέκταση αυτής της μεθόδου στην οποία η ωφελεία των παικτών αλλάζει σχετικά αργά στο χρόνο. Κάτι τέτοιο φαντάζει αρκετά λογικό αφού από μέρα σε μέρα δεν παρατηρούνται στην πράξη ριζικές αλλαγές στις ωφελείες. Πιο συγκεκριμένα, στο μοντέλο μας υποθέτουμε ότι οι παίχτες παίζουν δυναμικό no-regret και φτιάχνουμε ένα δυναμικό "λογικό" σύνολο, με όμοια λογική με αυτή στο στατικό μοντέλο. Δηλαδή, υποθέτουμε ότι οι παίχτες είναι αρκετά καλοί ώστε σε βάθος χρόνου να μπορούν να ανταγωνίζονται οποιαδήποτε ακολουθία διαφορετικών πονταρισμάτων. Για να μπορέσουν να παίξουν τόσο καλά οι παίχτες θα πρέπει να μην παρατηρούνται πολύ μεγάλες αλλαγές στο περιβάλλον των δημοπρασιών, αλλίως η διαδικασία της μάθησης δεν μπορεί να επιτευχθεί. Η συνάρτηση που προσπαθεί να βελτιστοποιήσει ο παίχτης έχει τη μορφή $u_t(b_t) = v_t P_t(b_t) - C_t(b_t)$. Παρατηρούμε ότι υπάρχουν τρεις "ποσότητες" που επηρεάζουν τη συνάρτηση, οι $v_t, P_t(b_t), C_t(b_t)$. Η πρώτη είναι η παράμετρος της ωφελείας που επιθυμούμε να μάθουμε και καθορίζεται από τον ίδιο τον παίχτη, ενώ οι άλλες δύο είναι η πιθανότητα κλικαρίσματος καθώς και το κόστος ανά κλικ, που καθορίζονται από τον μηχανισμό και το περιβάλλον του παίχτη. Το παρακάτω θεώρημα μας δείχνει το μέγεθος των αλλαγών που επιτρέπονται σε αυτές τις ποσότητες ώστε να επιτευχθεί η μάθηση.

Θεώρημα:Έστω $\epsilon_{P_t(x)} = P_t(x) - P_{t-1}(x), \epsilon_{C_t(x)} = C_t(x) - C_{t-1}(x)$. Αν $v_t \leq V, \forall t$, τότε για να έχει ο παίχτης δυναμικό regret που "εξαφανίζεται" στο πέρασμα του χρόνου θα πρέπει να ισχύουν οι ακόλουθες συνθήκες:
- $\sum_{t=2}^{T} |v_t - v_{t-1}| = o(T)$
- $\sum_{t=2}^{T} \sup_{x \in X} |\epsilon_{P_t(x)}| = o(T)$
- $\sum_{t=2}^{T} \sup_{x \in X} |\epsilon_{C_t(x)}| = o(T)$

Από το παραπάνω θεώρημα βλέπουμε ότι οι επιτρεπόμενες αλλαγές είναι πολύ μεγάλες. Δε θα μπορούσαμε να ελπίζουμε ότι ο παίχτης θα μάθει κάτι σε περίπτωση που το περιβάλλον αλλάζει τελείως από γύρο σε γύρο.

Η συνθήκη ότι ο παίχτης πετυχαίνει δυναμικό no-regret μεταφράζεται στην παρακάτω σχέση για τις ποσότητες που εξετάσαμε πριν.

$$\frac{1}{T}(\sum_{t=1}^{T} v_t P_t(b_t) - C_t(b_t)) \geq \frac{1}{T}(\sum_{t=1}^{T} v_t P_t(b'_t) - C_t(b'_t)) - \epsilon, \forall \boldsymbol{b'} \in B^T$$

Βλέπουμε ουσιαστικά ότι ο παίχτης παίζει τόσο καλά (κατά μέσο όρο) όσο οποιαδήποτε άλλη ακολουθία πονταρισμάτων, μείον κάποιο μικρή τιμή η οποία καθώς περνάει ο χρόνος πηγαίνει

στο 0. Αν ορίσουμε $\Delta P_t(b'_t) = \frac{1}{T}(P_t(b'_t) - P_t(b_t)), \Delta C_t(b'_t) = \frac{1}{T}(C_t(b'_t) - C_t(b_t))$, που μετράνε την (κανονικοποιημένη) αλλαγή στην πιθανοτήτα και στο κόστος του κλικ όταν στο γύρο t αλλάζει το ποντάρισμα από b_t σε σε b'_t, καθώς και $\boldsymbol{\Delta P(b')} = (\Delta P_1(b'_1), ..., \Delta P_T(b'_T)), \boldsymbol{\Delta C(b')} = (\Delta C_1(b'_1), ..., \Delta C_T(b'_T))$ η προηγούμενη συνθήκη της μάθησης γράφεται ισοδύναμα ως

$$\sum_{t=1}^{T} v_t \cdot \frac{1}{T}(P_t(b'_t) - P_t(b_t)) - \sum_{t=1}^{T} \frac{1}{T}(C_t(b'_t) - C_t(b_t)) \leq \epsilon, \forall \boldsymbol{b'} \in B^T$$

$$\boldsymbol{v \cdot \Delta P(b') - 1 \cdot \Delta C(b')} \leq \epsilon, \forall \boldsymbol{b'} \in B^T$$

Ορίζουμε ως δυναμικό "λογικό" σύνολο, το σύνολο των $(\boldsymbol{v}, \epsilon)$ που ικανοποιούν την προηγούμενη σχέση. Κάθε τέτοιο ζευγάρι μας λέει ότι αν ο παίχτης είχε την ακολουθία ωφελειών $\boldsymbol{v}$ τότε το δυναμικό του regret (με βάση το πώς έπαιξε) είναι το πολύ ϵ. Το σύνολο αυτό έχει πολλές καλές ιδιοτήτες, όπως αντίστοιχα είχε και στη "στατική" περίπτωση. Η πιο βασική απ' όλες είναι η ακόλουθη.

Θεώρημα: Το δυναμικό "λογικό" σύνολο είναι ένα κλειστό, κυρτό σύνολο.

Πρόβλεψη συγκεκριμένης ακολουθίας Το σύνολο που περιγράψαμε παραπάνω περιέχει ως στοιχεία του όλες τις πιθανές ακολουθίες που θα μπορούσε να έχει ο παίχτης, μαζί με το αντίστοιχο regret που αντιστοιχεί σε κάθε μία. Σε αντίθεση με το στατικό πρόβλημα που μελετήσαμε στην προήγουμενη ενότητα, σε αυτό το να απαντάμε απλά την ακολουθία που έχει το μικρότερο δυνατό δυναμικό regret δεν είναι τόσο καλό, για τον εξής λόγο. Σε κάθε χρονική στιγμή αφού μπορούμε να απαντάμε διαφορετική ωφελεία, μπορούμε να δίνουμε ως έξοδο την ωφελεία εκείνη που εξηγεί το ποντάρισμα αυτού του γύρου ως το βέλτιστο δυνατό. Με άλλα λόγια, αν απαντάμε απλά την ακολουθία με το ελάχιστο δυνατό regret το αποτέλεσμα θα είναι ότι τελικά θα υπάρχουν μεγάλες διαφορές από γύρο σε γύρο για να είναι μικρό το σφάλμα του παίχτη, πράγμα όμως που δεν είναι ρεαλιστικό. Παρακάτω παρουσιάζουμε το γραμμικό πρόγραμμα που απαντάει τη ζητούμενη ακολουθία, για να λύσει το πρόβλημα που αναφέραμε πριν.

$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{T} \epsilon_t \\ \text{subject to} \quad & \tfrac{1}{T}(v_t \Delta P(b') - \Delta C(b')) \leq \epsilon_t \ , t = 1, ..., T, b' = b_1, ..., b_{|B|} \\ & v_t - v_{t-1} \leq k, v_{t-1} - v_t \leq k, t = 2, ..., T \end{aligned}$$

Η λογική της απάντησης που δίνουμε είναι ότι ψάχνουμε μια ακολουθία από ωφελείες η οποία από τη μία έχει μικρό δυναμικό regret για τον παίχτη και από την αλλή έχει μικρές αλλαγές, ώστε να είναι πιο ρεαλιστική. Αυτό φαίνεται από τη δεύτερη συνθήκη του γραμμικού προγράμματος, όπου θέλουμε από γύρο σε γύρο οι ωφελείες να μη διαφέρουν περισσότερο από k. Βλέπουμε λοιπόν ότι για τη σωστή εκτίμηση της ακολουθίας υπάρχει ένας συμβιβασμός μεταξύ του πόσο μεγάλες αλλαγές ανεχόμαστε από γύρο σε γύρο και πόσο μικρό σφάλμα θέλουμε να έχει ο παίκτης.

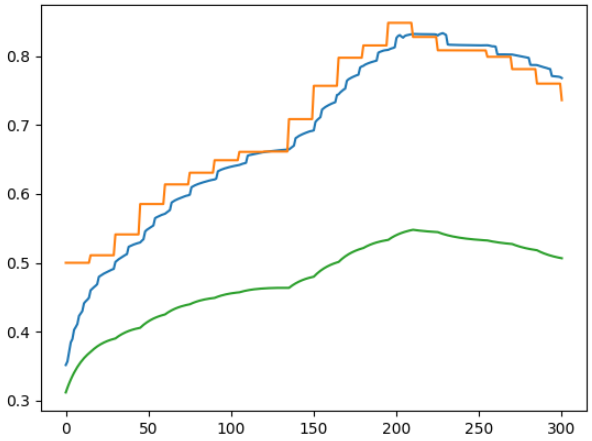
Εναλλακτική προσέγγιση στην πρόβλεψη ακολουθίας

Το πρόβλημα με την παραπάνω προσέγγιση είναι ότι οι υποθέτουμε ότι γνωρίζουμε εκ των προτέρων ένα φράγμα στο πόσο πολύ μπορούν να αλλάζουν οι ωφελείες από γύρο σε γύρο. Κάτι τέτοιο όμως πολλές φορές δεν μπορεί να είναι γνωστό, επομένως θα πρέπει να χρησιμοποιήσουμε μια εναλλακτική προσέγγιση που καλύπτει και αυτή τη περίπτωση. Αυτή η εναλλακτική προσέγγιση δίνεται από το παρακάτω γραμμικό πρόγραμμα.

$$\begin{aligned} \text{minimize} \quad & \lambda \sum_{t=1}^{T} \epsilon_t + (1-\lambda)k \\ \text{subject to} \quad & \tfrac{1}{T}(v_t \Delta P(b') - \Delta C(b')) \leq \epsilon_t \ , t = 1, ..., T, b' = b_1, ..., b_{|B|} \\ & v_t - v_{t-1} \leq k, v_{t-1} - v_t \leq k, t = 2, ..., T \end{aligned}$$

Όπως αναφέραμε και πριν υπάρχει ένας συμβιβασμός ανάμεσα στο πόσο μεγάλες αλλαγές επιτρέπουμε στην ακολουθία και πόσο μεγάλο σφάλμα αφήνουμε τον παίχτη να έχει. Αυτή η συνάρτηση ελαχιστοποίησης του γραμμικού προγράμματος δείχνει ακριβώς αυτό το συμβιβασμό. Θέλουμε δηλαδή να ελαχιστοποιήσουμε ταυτόχρονα το σφάλμα και τις αλλαγές στη συνάρτηση, ενώ το λ είναι μια παράμετρος που δείχνει πόσο βάρος θέλουμε να δώσουμε σε κάθε ένα από τους δύο όρους της ελαχιστοποίησης.

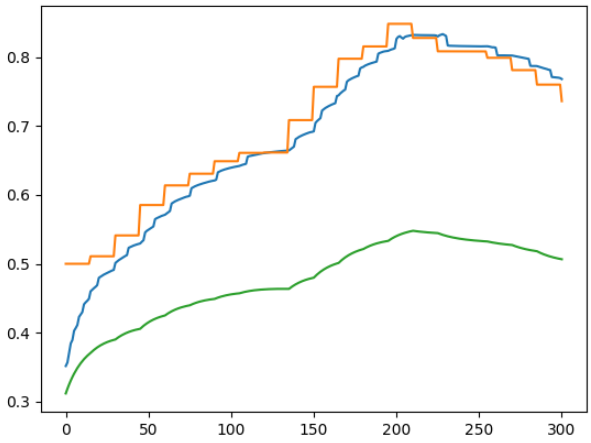
Παρακάτω φαίνονται κάποια αποτελέσματα προσομοιώσεων της μεθόδου μας από συνθετικά δεδομένα. Η πράσινη καμπύλη είναι οι προσφορές του παίχτη, η πορτοκαλί δείχνει τις πραγματικές ωφέλειες του παίχτη, ενώ η μπλε δείχνει την ακολουθία που δίνουμε ως απάντηση.



Σχήμα 1.2: Εξαγωγή Παραμέτρων Πρώτης Μεθόδου

Βελτιστοποίηση Κέρδους σε περιβάλλοντα ενός παίκτη-ενός αντικειμένου

Από τη σκοπιά του δημοπράτη, ενδιαφερόμαστε να χρησιμοποιήσουμε τη μέθοδο πρόβλεψης των ωφελειών ώστε να βελτιστοποιήσουμε το κέρδος μας. Επικεντρωνόμαστε σε περιβάλλοντα ενός παίχτη-ενός αντικειμένου όπου αυτή η παράμετρος της ωφελείας διατηρείται σταθερή στο χρόνο. Σε ένα τέτοιο περιβάλλον ο μόνος τρόπος με τον οποίο μπορεί ο δημοπράτης να επηρεάσει το κέρδος του είναι μέσω των τιμών κράτησης. Οι κανόνες του παιχνιδιού είναι απλοί: σε κάθε γύρο ο δημοπράτης πριν δει την προσφορά του παίχτη θέτει μία τιμή κράτησης και ο παίχτης (χωρίς να γνωρίζει την τιμή αυτή) υποβάλει μια προσφορά. Αν η προσφορά είναι μεγαλύτερη από την τιμή κράτησης τότε ο παίχτης παίρνει το αντικείμενο και πληρώνει ένα ποσό ίσο με την τιμή αυτή, ενώ σε διαφορετική περίπτωση δεν παίρνει και δεν πληρώνει τίποτα. Αυτό που θα κάνουμε είναι μια προσεκτική δυαδική αναζήτηση στο χώρο των ωφελειών, έχοντας ως αρχικό σημείο της αναζήτησής μας την τιμή ωφελείας που μας δίνει η μέθοδος πρόβλεψης. Η ιδέα είναι ότι αν το αποτέλεσμα της πρόβλεψης είναι κοντά στην πραγματική τιμή τότε θα μπορέσουμε γρήγορα να φτάσουμε όσο κοντά της θέλουμε. Ξεκινάμε με αυτή την τιμή και ανάλογα με το αν αγοραστεί ή όχι το αντικείμενο για $\sqrt{T}$ γύρους (ώστε να δώσουμε την ευκαιρία στον παίχτη



Σχήμα 1.3: Εξαγωγή Παραμέτρων Δεύτερης Μεθόδου

να μάθει να παίζει σε αυτή τη τιμή) αλλάζουμε την τιμή σε $v_0 + b_0, v_0 + 2b_0, v_0 + 4b_0, ...,$ ή $v_0 - b_0, v_0 - 2b_0, v_0 - 4b_0, ...,$ αντίστοιχα, όπου v_0 είναι το αρχικό σημείο και b_0 η ελάχιστη επιτρεπτή αύξηση της προσφοράς. Τέλος, μόλις αγοραστεί (ή δεν αγοραστεί) για πρώτη φορά το αντικείμενο κάνουμε μια δυαδική αναζήτηση στο τελικό μας διάστημα. Το παρακάτω θεώρημα δίνει το συνολικό κέρδος από αυτή τη διαδικασία.

Θεώρημα: Το συνολικό κέρδος από αυτή τη διαδικασία ανάθεσης τιμών είναι τουλάχιστον $(v^* - \epsilon)T - \Theta((\log \eta + \log \frac{\eta}{\epsilon})\sqrt{T})$, όπου T είναι οι συνολικοί γύροι του παιχνιδιού, η είναι η απόσταση της αρχικής πρόβλεψης από την πραγματική τιμή και ϵ είναι το πόσο κοντά στο πραγματικό v θέλουμε να φτάσουμε.

Chapter 2

Introduction

2.1 Motivation

It is undeniable that the emergence of the internet had a tremendous impact on marketing and especially advertising. Online advertising has, in a great extent, taken the place of traditional means of advertising such as newspapers, magazines and television. A very important type of online advertising are the so-called *sponsored search* ads that appear alongside the organic search results, when a user is searching for a specific keyword in some major search engine, such as Google, Yahoo! or Bing. These ads generate billions of revenue each year for the respective search engines, therefore it is of utmost importance for these companies to study them. There are major questions that arise not only from the search engines' perspective, but also from the advertisers' point of view. How should the search engines allocate the advertising spots and how should they charge the advertisers in order to maximize their profits? How should the advertisers behave to make the most of their advertising campaign? These are deep questions that need to be answered.

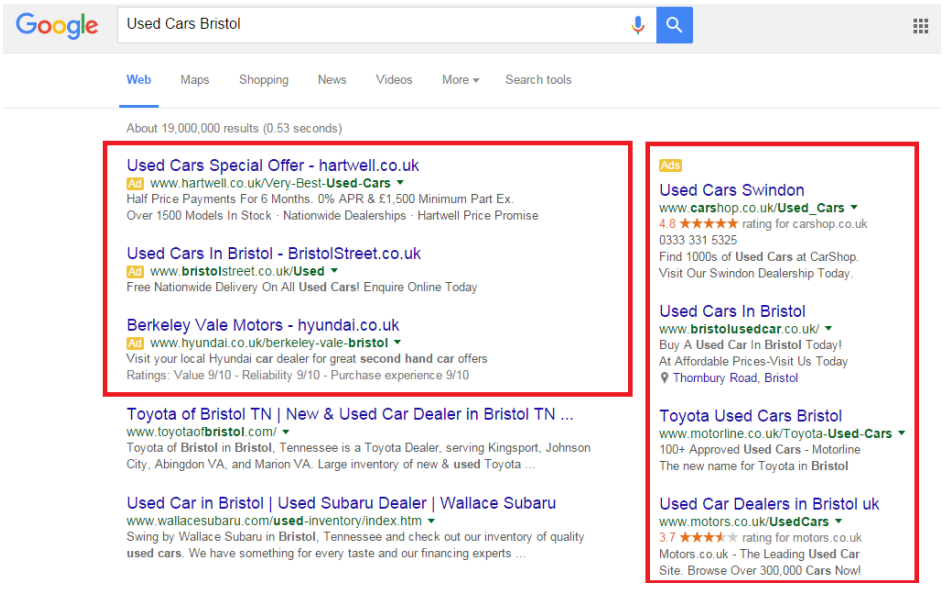


Figure 2.1: Sponsored Search Example

Before ads such as the ones that are shown above are displayed to the user, an auction takes place behind the scenes, which has very severe time constraints. The advertisers submit a bid and after the search engine has collected all the bids, it decides which slots to allocate at each advertiser (if any). The most prevalent auction format that is used in practice is the so-called *Generalized Second Price Auction*(GSP). There are also many platforms which use *First Price Auctions*. Roughly speaking, in the GSP mechanism each advertiser is associated

with a quality score, which is independent of the actual auction that takes place, and the bid that he submits. The mechanism uses the product of these two quantities in order to rank the bidders in descending order and allocates the k available slots to the first k players, as long as their score passes a predetermined quantity, which is unknown to them. Whenever an ad is clicked, the advertiser is charged the least amount he had to bid in order to maintain his ranking position. In the First Price auctions, the bidders are ranked according to the bids that they submitted, and after each click they are charged their actual bid. We will dig into the details of those mechanisms in the following chapter.

It turns out that in such auctions, although the format is pretty simple, it isn't easy, as a bidder, to decide on your bidding strategy. Almost all of the interesting mechanisms that are used in practice lack *truthfulness*; simply bidding how much they really "value" the item for sale is not the best strategy for the players. Under such mechanisms, the optimal behavior of the bidder depends heavily on his surrounding environment, which is the competition that he faces as well as some limitations that are imposed by the auctioneer. Therefore, agents should use some kind of machine learning techniques, that allow them to change their strategies dynamically in order to adapt to a contiuously varying environment. Data from major search engines confirm this belief, thus it is necessary to bear in mind this learning behavior when analyzing past data from those players. Nekipelov et al. [NST15] suggested that advertisers' behavior in such settings should be analyzed under the *online learning* framework. Roughly speaking, if we view each auction as a round of some game, this learning behavior allows the advertisers to have the same profit (on average) as if they had known all the rounds in advance and they were submitting the same bid at each round. We will give a detailed description of the online learning framework in Chapter 4.

Econometrics is a branch of economics which applies mathematical models to extract information from data regarding economic activities, in order to possibly forecast future trends from historical data or test the validity of some theory. A parameter which is of interest in the setting described above is the advertisers' valuation, which translates to their "happiness" for getting clicked. Knowledge of this paratemer allows the search engines to optimize the configuration of their auction formats, in order to maximize their revenue. When the mechanism that is used is truthful, retrieving the bidders' valuation is a trivial task; since bidding the true value is a dominant strategy past bids correspond to the actual valuation. However, things get more complicated when mechanisms are not truthful, thus it is necessary to make some further assumptions which are related to the state of the game.

2.2 Problem Statement

The general question that we are trying to answer is the following; Given past data from strategic interactions between players, how can we use them in order to estimate private parameters that are not observed? We focus our attention on repeated auctions that are not truthful and the participants are characterized by a single dimensional private valuation. We assume that players are learning agents and that their environment as well as their valuation vary slowly enough, so that they are able to learn how to behave under those new circumstances. The objective that each player i is trying to maximize every round t is $u_{it}(b_{it}) = v_{it} \cdot x_{it}(b_{it}) - c_{it}(b_{it})$, where b_{it} is the bid that he submits in that round, v_{it} is his private parameter for that round, $x_{it}(\cdot)$ is the *allocation function* for that round (it can be thought of as the probability of getting clicked in the sponsored search auction context) and $c_{it}(\cdot)$ is the function that determines the amount that the bidder is charged. Since we are viewing this problem from the auctioneer's perspective, for each player i and every round t we observe $b_{it}, x_{it}(\cdot), c_{it}(\cdot)$. The unknown parameter that we are trying to infer is v_{it} and we

want to do that for all the rounds and all the players that participate in the game.

2.3 Related Work

Classical econometric approaches that aimed to infer these parameters assumed that each advertiser's behavior had reached a stable state, which is known as a *Nash equilibrium* (see e.g. [AN10, BHN13, JLB07]). However, Daskalakis et al. [DGP09] showed that such an assumption is unrealistic, since it is computationally difficult for the players who interact in that way to reach such a state. Moreover, the amount of information that needs to be exchanges is not available to the players in such settings. Recently, Nekipelov et al. [NST15] proposed an alternative approach to that econometric problem which is based on the learning behavior that we explained before. Instead of assuming that the players have reached an equilibrium state in which each one best responds to his opponents strategies, they suggest that the strategies of the players are contiously changing in a way that minimizes their average regret. This assumption is much weaker and more realistic than that of Nash equilibria. Their method can be used in *single parameter* environments, which we will define formally in the next chapter, and capture many real-world scenarios. There is an ongoing line of work in algorithmic game theory literature which tries to characterize the effects that no-regret learning has on outcomes of games, such as approximate efficiency with respect to the optimal welfare (i.e the happiness of the people that participate in the auction) and revenue (i.e. the income that the auctioneer collects). For instance, Caragiannis et al. [CKKK11] proved that in the Generalized Second Price auction the average welfare of any no regret learning outcome is at least 30% of the optimal welfare. Syrgkanis et al. [SALS15] showed that when players use learning algorithms from a specific class, the welfare of the game converges to the optimal at rate $O(1/T)$ and each player's regret drops at rate $O(T^{-\frac{3}{4}})$, which is faster than the convergence rate of vanilla learning algorithms. Foster et al. [FLL+16] showed that a wider class of algorithms can achieve these results, with less information than it was required before. Lykouris et al. [LST16] explored a dynamic setting, in the sense that the players that participate in the game change over time. They showed that in many games, even when there are significant variations from round to round, the welfare of the game is close to the optimal. Braverman et al. [BMSW18] analyzed single item-single buyer auctions under learning assumptions and provided a fairly complete characterization for that setting, showing that there are some cases in which the seller can extract revenue arbitrarily close to the buyer's valuation. We will discuss these results further in chapter 5.

2.4 Contribution

In this thesis we extend the inference method proposed by Nekipelov et al. [NST15], dropping the assumption that the advertisers' valuation, which is their "happiness" for getting clicked by the user, remains constant over time. The challenge in that setting is that instead of answering just a scalar value we have to answer a sequence of them, so the search space gets very large quickly and we have to determine which of these sequences are meaningful. In our model, we allow the advertisers' valuation to vary slowly over time and we propose a method which estimates the sequence of these changing valuations. We assume that bidders are *dynamic* no regret learners, which means that their performance is (on average) as good as the performance of any other bidding sequence they could have chosen. Although this assumption might seem very strong, we show that in order to achieve that, the deviation of their valuations sequence can be up to $o(T)$. We present an experimental evaluation of the aforementioned method which shows some very encouraging results, since the predicted

sequence of valuations is very close to the actual one. This method is independent of the underlying mechanism through which the advertisng slots are allocated and can be used to infer the valuations of players in any single-parameter environment. Furthermore, we develop a pricing method in the *single item-single buyer* setting for a *no regret* buyer, whose valuation remains constant over time, using the [NST15] inference method. We set the initial selling price to the predicted valuation and continue by doing a binary search on the valuation space. Since the valuation remains constant, we can get (on average) revenue that is arbitrarily close to the valuation, as expected. We show that the convergence rate of our pricing method depends on the distance between the predicted and the actual valuation.

2.5 Organization

• The **first** chapter serves as an extended summary of the thesis in greek.

• The **second** chapter is a general overview of the problem domain that was investigated.

• In the **third** chapter we introduce the reader to the basic concepts of Mechanism Design, which aims to design systems that are to be used by strategic agents. We define some formal models for the auctions that we are interested in and present some important results from that area.

• In the **fourth** chapter we formalize the *online learning model* that we discussed before. We start by presenting algorithms that achieve the learning goal in discrete environments and continue by describing algorithms for the continuous setting. The main purpose of this chapter is to show how bidders who participate in auctions can behave in order to maximize their profit, but the setting is actually very general and can model many other problems.

• In the **fifth** chapter we present some important results from the ongoing line of work which tries to analyze and understand these type of auctions under a *no regret learning* point of view.

• Finally, in the **sixth** chapter we present our results, which were shortly discussed earlier.

Chapter 3

Equilibrium Concepts and Mechanism Design

Game theory studies the interactions between strategic agents, who act in a selfish way in order to maximize their own objectives. A very important notion in game theory is that of an *equilibrium*. Equilibria are states in such interactions in which the players have no reason to unilaterally deviate from their current strategies, therefore equilibria are steady states in the sense that if the agents somehow manage to reach them, it is reasonable to believe that their strategies will remain the same. A particularly interesting steady state arises when players have *dominant strategies*, meaning that no matter what their opponents are doing, it is in their best interest to behave in a specific way. Thus, when games are designed in a way that provide the players with dominant strategies it is very easy to predict the outcome.

Mechanism design is the science of designing rules that regulate the interaction of such strategic entities. Those entities can be people who participate in an auction, drivers who wish to arrive somewhere as soon as possible, sports teams that participate in tournaments, advertisers who wish to display their ads after a search for a specific keyword, etc.. It is very important, and sometimes difficult, to design systems that will yield a "good" result, no matter what the preferences of the participating agents are. We will define later two notions of "good" results, which depend on the system's desginer goals. Ideally, we would like to have mechanisms that provide agents with dominant strategies, which are efficently computable.

As an introductory mechanism design problem, imagine that we want to give away an item that is of no use to us anymore, but we are generous enough to not care about what we will get in return, we just want to give the item to the person that wants it the most. A first approach would likely be to just ask all the potential receivers of that item to write down a number that represents "how much" they want it, and give it to the one with the biggest number. The problem is that since those receivers are *strategic*, they have no reason at all to tell us the truth; in fact, their best strategy is to write the largest number that fits in their piece of paper. Clearly, the previous mechanism won't achieve the goal that we had in mind, it is an example of a poorly designed system. What can we do to fix that?

3.1 Equilibrium Concepts

As we briefly discussed above, in strategic interactions between players such as auctions, there are some "stable" situations which enjoy various interesting properties and can help us analyze these games. Let's now define a "game" formally, in order to define some of the equilibrium concepts. The interaction we are interested in is called a *Cost Minimization Game* and consists of the following elements.

Definition 3.1.1. Cost Minimization Game

- n players, where n is a finite number
- a finite *strategy space* S_i for each player i. In the auctions setting, this strategy space consists of the different bids that each player is allowed to submit. We denote by $\boldsymbol{s} = (s_1, ..., s_n)$ the *strategy profile* of the game
- a *cost function* $C_i : S_1 \times ... \times S_n \to R$ for each player i (each bidder wants to minimize his cost)

We will present five important *equilibrium* concepts, which are different notions of stable situations.

Definition 3.1.2. Dominant Strategy
A strategy s' for player i is dominant if $\forall s \in S_i, \forall \boldsymbol{s}_{-i} \in S_1 \times ... \times S_{i-1} \times S_{i+1} \times ... \times S_n$ we have that

$$C_i(s', \boldsymbol{s}_{-i}) \leq C_i(s, \boldsymbol{s}_{-i})$$

Definition 3.1.3. Pure Nash Equilibrium-PNE
A strategy profile $\boldsymbol{s}$ is a pure Nash equilibrium if for every player i and every unilateral deviation $s_i' \in S_i$ we have that

$$C_i(\boldsymbol{s}) \leq C_i(s_i', \boldsymbol{s_{-i}})$$

If every player i has a dominant strategy it is very easy for the player to participate in it and for the designer to predict its outcome. Unfortunately not all games have dominant strategies, therefore we have to generalize the equilibrium concepts.

This definition is due to Nash [Nas51]. Notice that a pure Nash equilibrium is a weaker notion than a dominant strategy, since it only requires that each player best responds to some specific strategies that the others follow, and not that he should behave in a specific way, no matter what the others are doing. Unfortunately, not every game has a pure Nash equilibrium, so we will extend that definition a little bit in order to obtain universality.

Definition 3.1.4. Mixed Nash Equilibrium-MNE
Distributions $\sigma_1, ..., \sigma_n$ over $S_1, ..., S_n$ respectively constitute a mixed Nash equilibrium if for every player i and every unilateral deviation $s_i' \in S_i$ we have that

$$\mathbb{E}_{\boldsymbol{s} \sim \sigma}[C_i(\boldsymbol{s})] \leq \mathbb{E}_{\boldsymbol{s_{-i}} \sim \sigma_{-i}}[C_i(s_i', \boldsymbol{s_{-i}})]$$

where by σ we denote the product distribution $\sigma_1 \times ... \times \sigma_n$.

This definition is a generalization of the previous in the sense that it permits each player to randomize over his strategies, so if we set the distributions to be a fixed action, we can obtain the previous. Nash [Nas51] proved that each game has at least one mixed Nash equilibrium, so under this new definition we have obtained the universality that we looked for. Unfortunately, Daskalakis et al. [DGP09] proved that in general games it is hard to compute mixed Nash equilibria. So we have to extent our definitions even more in order to find tractable notions.

The intuition behind the next equilibrium concept that we will define is the following. Suppose that every player knows a common joint distribution σ and that there is a trusted third party which draws samples from that distribution and reports them to the players. We want that distribution to have the property that after the third party reveals some strategy s_i to player i, then assuming that everyone else follows their suggestion, i has no reason to deviate. An example of a correlated equilibrium is the traffic light. We know that there is

distribution σ from which the traffic light draws its proposals, such that when we are shown the green light the other cars are shown the red and vice versa. In a situation like that, our best move is to simply follow the strategy that the light indicates. Let's now define the notion of a *correlated equilibrium* formally.

Definition 3.1.5. Correlated Equilibrium-CE
A distribution σ over $S_1, ..., S_n$ is a correlated equilibrium of a cost minimization game if for every player i and every unilateral deviation $s_i' \in S_i$ we have that

$$\mathbb{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})|s_i] \leq \mathbb{E}_{\boldsymbol{s_{-i}}\sim\sigma_{-i}}[C_i(s_i', \boldsymbol{s_{-i}})|s_i]$$

By generalizing our definition, we now have equilibria that are tractable. Correlated equilibria are also guaranteed to exist. Notice that we didn't require σ to be a product distribution, so the strategies of the players can be arbitrarily correlated. Also notice that player i doesn't know the suggestion that the trusted authority made to the other players, he just knows his own suggestion and the distribution σ. In the following chapter we will discuss algorithms that reach such a stable state.

We will generalize our definition a little further and introduce the so-called *coarse correlated equilibria*. This time, we only require that the distribution is known to the players, and that the players follow the suggested strategy by the trusted third party without even seeing it first.

Definition 3.1.6. Coarse Correlated Equilibrium-CCE
A distribution σ over $S_1, ..., S_n$ is a correlated equilibrium of a cost minimization game if for every player i and every unilateral deviation $s_i' \in S_i$ we have that

$$\mathbb{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})] \leq \mathbb{E}_{\boldsymbol{s_{-i}}\sim\sigma_{-i}}[C_i(s_i', \boldsymbol{s_{-i}})]$$

We can see that a CCE protects agains unconditional unilateral deviations, whereas the CE protects against conditional. Every CE is also a CCE, so CCEs are also guaranteed to exist in every finite game and are tractable as well. The reason why we extended the notion of CE to CCE is that there are learning algorithms that reach a CCE which are simpler and more natural compared to those that achieve a CE.

It is illustrative to depict the notions we defined above.

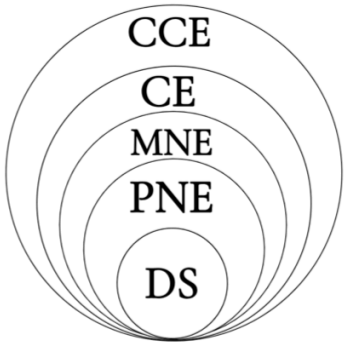


Figure 3.1: Hierarchy of equilibrium concepts

3.2 Single Item Auction Model

Let us now define a formal model in the so-called *single item auction* setting, which we will use throughout the text, and will help us tackle the problem discussed in the introduction. We assume that there is a single item for sale by an auctioneer, n potential buyers (or *players*), and each player i has a private *valuation* v_i, which represents "how much" he wants that item, and it is measured in monetary units (we don't care *how* the buyer actually arrives at this value, we just assume that he does). This value can also be interpreted as the maximum amount that he is willing to pay for that item. The auctioneer is responsible for determining who gets the item, if any, and how much he is charged for that. That payment is denoted by p. We assume that the overall "happiness" of each participant i, called *utility* and denoted by u_i, is quasilinear, namely that

$$u_i = \begin{cases} v_i - p, & \text{if } i \text{ receives the item} \\ 0, & \text{otherwise} \end{cases}$$

The auction, sometimes referred to as *game*, proceeds in the following way.

1. Each player i, simultaneously and privately, reports his bid b_i to the auctioneer.
2. The auctioneer, based on these bids, decides who gets the item.
3. The auctioneer decides the amount that the winner is charged.

Our goal as auctioneers is to design allocation rules (i.e. rules that dictate who gets the item) and payment rules that incentivize the players to act in a *predictable* way, so that we can reason about the outcome of the auction. Even if we are altruists and don't care about the money that we will get in return, payment rules are necessary to avoid the situation that was described in the introductory example.

Formally, in the single item setting, B is the set of all possible bids (we assume that it's the same for each player). The bids that were collected constitute a *bid profile*, $\boldsymbol{b} = (b_1, ..., b_n)$. The allocation rule $\boldsymbol{x(b)} : B^n \to \{0,1\}^n, s.t. \sum_{i=1}^n x_i \leq 1$, is a mapping from the bid profile to the winner of the item, namely

$$x_i = \begin{cases} 1, & \text{if } i \text{ receives the item} \\ 0, & \text{otherwise} \end{cases}$$

The payment rule, $p(\boldsymbol{b}) : B^n \to \mathbb{R}$, is a mapping from the bid profile to the real numbers, that corresponds to the amount that the winner is charged. For simplicity, we will overload the notation and sometimes omit the dependence of both $\boldsymbol{x}, p$ on $\boldsymbol{b}$. It is clear that each player's utility depends on the allocation $\boldsymbol{x}$ and payment p, which depend on $\boldsymbol{b}$, and to emphasize that dependece we will sometimes denote it as $u_i(\boldsymbol{b})$.

Under those definitions the two objectives that we will consider, which measure the success of our mechanisms, are the *Social Welfare* and the *Revenue*. The Social Welfare of an allocation $\boldsymbol{x}$, $W(\boldsymbol{x})$, is defined as $W(\boldsymbol{x}) = \sum_{i=1}^n v_i \cdot x_i$, whereas the Revenue is simply $Rev(\boldsymbol{x}) = \sum_{i=1}^n p \cdot x_i$.

In the following sections we will focus our attention on auction formats that maximize the Social Welfare objective.

3.2.1 First Price Auction

A very natural allocation rule for the problem described above is to just give the item to the highest bidder, whereas the payment rule is to simply charge him the amount that he decided to bid. Although from the auctioneer's perspective this auction is very simple (and very fast to implement), it is not easy to find out how you should play as a bidder. Clearly a bidder has no reason to bid his true valuation v, because, in that way, his utility will always be 0. Therefore, there is an inherent tradeoff between bidding near his valuation and possibly paying more than needed and bidding far below his valuation, with the risk of losing the item. In order to decide his best strategy, the bidder has to make some assumptions about his competition, namely to find out the number of bidders that participate in the auction, their valuations' distribution and how these are correlated between them. Hoy et al. [HTW18] proved that under those assumptions, assuming furthermore that valuation distributions are independent, the Welfare of the First Price Auction is at least an .743 approximation of the optimal Welfare, whereas Syrgkanis [ST13] proved that this mechanism gives an $\frac{e-1}{e}$ approximation, which is tight when the valuations are correlated.

The previous mechanism shows us that we should try to design systems that make it easy for the participants to decide on their strategy, no matter what the other palyers do. We want rules that provide agents with *dominant strategies*, i.e. bidding decisions which depend solely on their valuation. We also want these dominant strategies to be efficiently computable. The First Price Auction lacks the dominant strategy that we seek.

3.2.2 Second Price Auction

The Second Price Auction, or Vickrey Auction, proposed by Vickrey [Vic61], has the same allocation rule as the First Price Auction, but the payment is different; this time the winner has to pay the second highest bid and not his own. The motivation behind that payment rule is that we should charge the winner just the amount that he had to bid in order to win the item. This simple modfication in the previous mechanism has very important consequences. More precisely, it is now clear what every player should do; they should simply report their true valuation. This auction has a very simple dominant strategy, which is proved formally below.

***Theorem* 3.2.1.** *In the Second Price Auction every bidder i has a dominant strategy, to set $b_i = v_i$, meaning that for any fixed $\boldsymbol{b}_{-i} \in B^{n-1}$ it holds that $u_i(v_i, \boldsymbol{b_{-i}}) \geq u_i(b, \boldsymbol{b_{-i}}), \forall b \in B$.*

Proof. Let $b^* = max_{j \neq i} b_j$ be the highest bid among the other players. From the auction format, if $b_i < b^* \implies u_i = 0$, whereas if $b_i \geq b^* \implies u_i = v_i - b^*$. Consider two cases for player i. If $v_i < b^*$ then underbidding still leads to 0 utility, while overbidding leads to negative utility because the payment will be higher than the valuation. If $v_i >= b^*$ then overbidding makes no difference as the player i still wins the item and pays b^*, whereas underbidding might lead to some other player $j \neq i$ getting the item, leaving i with 0 utility.
Therefore, setting $b_i = v_i \implies u_i = \max\{0, v_i - b^*\}$

□

We call mechanisms in which telling the truth is a dominant strategy *truthful*. A corollary that follows immediately from the previous theorem is that the Second Price Auction maximizes the Social Welfare. If we assume that the bidders are *rational*, i.e. they follow their dominant strategy, then the mechanism allocates the item to the player that wants it the most. Moreover, everyone's utility is non-negative, so no player regrets participating in the auction, even if they don't win the item.

The Second Price Auction is a very good mechanism because it has the following three properties.

1. It is a dominant strategy for every player to report his true valuation (**incentive guarantees**).
2. If players act rationally it maximizes a predefined objective, in that case the Social Welfare (**perfomance guarantees**).
3. It runs in polynomial, more precisely linear, time (**computational efficiency**).

3.3 Myerson's Lemma

In this section we will generalize the model that we defined earlier a little bit. We can imagine this new environment as having an item which is divisible and we can sell a portion of it to each bidder.

Definition 3.3.1. Single Parameter Environment

- Every player i has a valuation $v_i \in \mathbb{R}$ which represents his "happiness" for each unit of item that he gets.
- Every player i reports a bid $b_i \in \mathbb{R}$ to the mechanism.
- There exists a feasible set X which enforces some restrictions on the mechanism and depends on the specific application. For instance in the Single Item Auction $X = \{(x_1, ..., x_n) | x_i \in \{0, 1\}, \sum_{i=1}^{n} \leq 1\}$. If we have an auction with k identical goods then $X = \{(x_1, ..., x_n) | x_i \in \{0, 1\}, \sum_{i=1}^{n} \leq k\}$.
- There is an allocation rule $\boldsymbol{x(b)} : B^n \to X$, in the same way as before.
- There is a payment rule $\boldsymbol{p(b)} : B^n \to \mathbb{R}^n$. Notice that now the payment is a vector instead of a real number.
- The utility of each bidder i is $u_i(\boldsymbol{b}) = v_i \cdot x_i(\boldsymbol{b}) - p_i(\boldsymbol{b})$.

The rules of the game remain the same as before. The mechanism collects the bids privately, chooses a feasible allocation and then charges each player a specific amount. A natural restriction for the payments is that $p_i(\boldsymbol{b}) \in [0, b_i \cdot x_i(\boldsymbol{b})]$. The reason for the first restriction is that it doesn't make sense for the auctioneer to pay the buyers, while the second ensures that if the bidders play truthfully then their utility will be non-negative. Before stating the actual theorem we need two more definitions.

Definition 3.3.2. Implementable Allocation Rule
An Allocation Rule $\boldsymbol{x}$ for the single parameter environment defined above is *implementable* if there exists a payment rule $\boldsymbol{p}$, such that bidding truthfully in the $(\boldsymbol{x}, \boldsymbol{p})$ mechanism is a dominant strategy.

Definition 3.3.3. Monotone Allocation Rule
An Allocation Rule $\boldsymbol{x}$ for the single parameter environment defined above is *monotone* if for every bidder i and every $\boldsymbol{b_{-i}}$ bid profile the function $\boldsymbol{x}(b, \boldsymbol{b_{-i}})$ is non-decreasing in b.

Myerson's lemma [Mye81] consists of three parts and provides a complete characterization of the mechanisms that achieve the goals we had in mind in the single parameter environment. More precisely, it provides a (simple) necessary and sufficient condition for an allocation rule to be *implementable* and gives an exact formula for the payments that this mechanism must impose. We state it formally below.

***Theorem* 3.3.1.** *In the single parameter environment the following hold*

1. *An allocation rule $\boldsymbol{x}$ is implementable if and only if it is monotone.*
2. *If the allocation rule is monotone then there exists a unique payment rule $\boldsymbol{p}$ such that the mechanism $(\boldsymbol{x}, \boldsymbol{p})$ is truthful.*
3. *The formula for that payment rule is $p_i(b_i, \boldsymbol{b_{-i}}) = \int_0^{b_i} z \cdot \frac{d}{dz} x_i(z, \boldsymbol{b_{-i}}) dz$.*

Myerson's Lemma provides a very strong result for the not-so-trivial single parameter environment. It states that whenever we want to design a truthful mechanism for such an environment we just have to make sure that the allocation rule is monotone, we don't have to think about the payments at all. Moreover, no matter how hard we try, we cannot design a truthful mechanism in which the allocation rule isn't monotone. An immediate corollary is that the single item auction that we analyzed before has a unique truthful mechanism; the Second Price Auction.

3.4 Revenue Maximizing Auctions

In this section we will switch gears and we will examine mechanisms that care about a different objective instead of the Social Welfare, namely the Revenue. In the mechanisms that we examined before, the only reason that we had to impose payments on the buyers was to incentivize them to bid their true valuation, therefore although there was revenue generated for the auctioneer, it was only a as side effect of some other goal. We investigated auctions which achieve the previous goal for two main reasons; First, there are many real-world scenarios in which maximizing the Social Welfare is the actual goal of the auctioneer, for instance government auctions for wireless sprectrum. The other important reason is that Welfare maximizing auctions in the single parameter environment can achieve their goal without any prior information about the participants, and they can be implemented in polynomial time - they seem to be able to handle strategic interactions for free. In contrast, Revenue maximizing mechanisms can't be achieved without some prior knowledge about the players that they are dealing with.

As a motivating example consider an auctioneer who wants to sell a single item to a single buyer. If the auctioneer wants to maximize the Social Welfare his job is trivial, he just has to give it to him for free. The selling price is the same for all potential buyers. Consider now what happens when the auctioneer wants to maximize his revenue instead; He definitely cannot give the item for free, his optimal strategy would be to price it ϵ below the buyer's valuation. The problem is that this valuation is unknown to the auctioneer, therefore in order to approach the problem we have to make some assumptions about the valuation, even though we don't know its exact value. Even in a simple instance of the Revenue maximization problem we can see that the seller has to behave differently when dealing with different buyers. Below we define a formal model under which we present a Revenue maximization mechanism.

Definition 3.4.1. Bayesian Model for Revenue Maximization

- We have a single parameter environment.
- We assume that each bidder i draws his valuation from a known distribution F_i, with a density function f_i which has bounded support. We also assume that distributions $F_1, ..., F_n$ are independent. These distributions encode some prior knowledge that the auctioneer has about the participants. Notice that we only require that the distributions of the valuations are known, but not the realizations of these distributions. Information like that can be obtained from historical data.
- The bidders don't need to know the distributions.

In the model that we described above, our goal is to find a truthful mechanism that achieves the highest possible revenue *in expectation*. For instance, in the example that we discussed above the optimal pricing strategy is $r = arg\max r(1 - F(r))$, where r is the revenue generated by the sale, whereas $1 - F(r)$ is the probability of actually selling the item. Below we state a theorem by Myerson [Mye81] which characterizes optimal auctions in the environment we defined above.

***Theorem* 3.4.1.** *Let F be the joint valuation distribution function and $\phi_i(v_i) = v_i - \frac{1-F_i(v_i)}{f_i(v_i)}$. Then $\mathbb{E}_{\boldsymbol{v}\sim F}[\sum_{i=1}^{n} p_i(\boldsymbol{v})] = \mathbb{E}_{\boldsymbol{v}\sim F}[\sum_{i=1}^{n} \phi_i(v_i) \cdot x_i(\boldsymbol{v})]$.*

We can prove the above theorem by expressing $\boldsymbol{p}$ using Myerson's formula and then manipulating the integrals. The terms $\phi_i(v_i)$ are called virtual valuations. What it actally tells us is that maximizing the expected revenue boils down to maximizing the expected *virtual* social welfare. In order to do that, we will maximize the virtual social welfare function pointwise, meaning that for every input $\boldsymbol{v}$ we will choose an $\boldsymbol{x}$ which maximizes $\sum_{i=1}^{n} \phi_i(v_i) \cdot x_i(\boldsymbol{v})$. Let's consider now the single item auction. Should we just allocate the item to the bidder who has the highest virtual valuation? We have to bear in mind that virtual valuations can be negative, so there are some instances in which the best thing to do is not give the item at all. Remember that we are interested in designing payments in a way that induce truthful auctions, so we can safely assume that bids correspond to actual valuations. In order to find out if our previous virtual welfare maximizing rule can be extended to a truthful mechanism we have to ask if it is monotone. The monotonicity of this rule depends on the underlying distribution function, whenever $\phi_i(v_i)$ is increasing in v_i the allocation rule defined above is monotone. Distributions F for which the virtual valuation function is monotone are called *regular*

Let's return again to the single item case, assuming further that bidders are i.i.d. and that their distributions are regular. Since $\phi_i(\cdot)$ is the same for all bidders, we can drop the dependence on i. In that case, the allocation rule described above is to simply give the item to the bidder with the highest virtual bid, which in that case corresponds to the bidder with the highest bid, as long as it is higher than a predefined value, which is $\phi^{-1}(0)$. If i is the winner, the payment is $p = \max\{\max_{j\neq i} b_j, \phi^{-1}(0)\}$. Therefore, we can view $\phi^{-1}(0)$ as an extra bidder who bids in favor of the auctioneer. Then, the Revenue maximizing auction is simply a Second Price Auction with that extra bidder. The value $\phi^{-1}(0)$ is called *reserve price*.

3.5 Online Ad Auctions

As we mentioned in the introductory chapter, the most dominant way to sell advertising spots in online keyword auctions is the GSP mechanism. To show the importance of

this mechanism, it is worth mentioning that a large fraction of Google's and Yahoo's revenue came from sponsored search auctions. More precisely, in 2005 over 98% of Google's 6.14 billions revenue was generated from this type of auctions, while in the same year it is believed that over half of Yahoo's 5.26 billions revenue came from the same source. For a more recent analysis of Google's advertising revenues we refer the interested reader to https://www.wordstream.com/articles/google-earnings.

We will start by providing a high level description of the system these search engines use, the GSP mechanism. Each time a user searchs for some specific keywords an auction takes place, which has to be executed in real time. Then, alongside the search engine's results, which are referred to as *organic results*, some ads could be displayed. In order for that ad to be displayed, the advertising company submits a bid to the search engine, which represents the maximum amount that they are willing to pay if their ad gets clicked by the user. So whenever an ad is getting clicked by a user, the respective advertiser has to pay an amount which is less or equal to the bid they submitted. It makes sense to use the Single Parameter Model that we described in the previous section to model the situation at hand; the advertiser has a private valuation v which represents their "happiness" for getting clicked. This valuation corresponds to the expected profit that the advertiser gets when a user visits their site. We will use the traditional *seperable click-through rates* model, which states that the probability of an ad getting clicked is simply the product of two factors, the advertiser's probability and the ad's position. We note that there are more expressive models that might be more realistic, such as the Cascade Click Model by Kempe and Mahdian [KM08]. Below we define the model formally.

3.5.1 Model

It is worth mentioning that each bidder i is associated with a factor γ_i, which represents the bidder's click probability, and a scoring factor s_i which represents their quality. Every advertising position j is associated with a factor a_j, which is independent of the ad that will eventually get that place. So in the seperable click-through rate model, if bidder i is awarded the position j his click probability is simply $a_i \cdot \gamma_j$. We have the following elements as part of the system:

1. n bidders that participate in the auction
2. m advertsing positions that can be sold
3. a private parameter v_i for each bidder, as we discussed above
4. $\boldsymbol{a} = (a_1, ..., a_m)$, a vector of position coefficients that are assumed to be non decreasing, i.e. $a_1 \geq a_2 \geq ... \geq a_m$, since in that model it doesn't make sense to hurt the bidder if his ad is displayed higher
5. $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_n)$, a vector of bidders' click probabilities
6. $\boldsymbol{s} = (s_1, ..., s_n)$, a vector of bidders' scoring coefficients
7. r, the reserve score of the auction. If a player's rank-score (which we'll define shortly) is lower than that, then his ad won't be displayed no matter what the other bidders do
8. $\boldsymbol{b} = (b_1, ..., b_n)$ which represents the bids that were submitted by the players

Let's see now how the game proceeds. Every advertiser is asked to submit a bid b_i to the system. Then, bidders are ranked by their so-called *rank score* $q_i = s_i \cdot b_i$, and are allocated the m positions in decreasing order as long as they pass the reserve score of the auction. The reason why the search engines use the ranking coefficients is that they are only getting paid after the user clicks on some ad. Therefore, even if a player bids very highly, if users never click his ads, because his products or services are of low quality, the search engine's revenue will be zero, so these coefficients can help solve this problem. If advertiser i is allocated the position j his click probability is $p_{ij} = a_j \cdot \gamma_i$ (*click through rate*-CTR). When that advertiser is clicked, his payment (*cost per click*-CPC) is the minimum bid he had to submit in order to keep his position, namely $c_{ij}(\boldsymbol{b}, \boldsymbol{s}, r) = \max\{\frac{s_{\pi(j+1)} \cdot b_{\pi(j+1)}}{s_i}, \frac{r}{s_i}\}$, where by π_j we denote the advertiser that is allocated position j. The allocation and the payment rules are the reason that this mechanism is called the *Generalized Second Price* (GSP) auction. Let $P_i(\boldsymbol{b}) = a_{\sigma_i(\boldsymbol{b})} \cdot \gamma_i$, where $\sigma_i(\boldsymbol{b})$ is the slot that i is allocated under bid profile $\boldsymbol{b}$. Let $C_i(\boldsymbol{b}) = a_{\sigma_i(\boldsymbol{b})} \cdot \gamma_i \cdot c_{i\sigma_i(\boldsymbol{b})}(\boldsymbol{b})$, be the expected payment of i. Then, we assume that his utility is $u_i(b_i) = v_i \cdot P_i(\boldsymbol{b}) - C_i(\boldsymbol{b})$.

The next question follows immediately after the description of the GSP mechanism. Is it truthful or could it be better for the players to report a different bid than their true valuation? The answer is negative and it's illustrative to see an example in which underbidding is a strictly better strategy than telling the truth. Imagine a setting with three bidders and two slots, where $\gamma_i = \sigma_i = 1, \forall i$, $a_1 = 1, a_2 = 0.4, r = 0$ and $v_1 = 7, v_2 = 6, v_3 = 1$. Let's see what happens for player 1 when players 2,3 bid truthfully. If he plays truthfully as well, he will earn the first slot, therefore his expected utility will be $(v_1 - b_2) \cdot a_1 = 1$. If he underbids, setting $b_1 = 5$ for example, he will earn the second slot and his utility will be $(v_1 - b_3) \cdot a_2 = 6 \cdot 0.4 = 2.4$. Therefore, even in this trivial example we can see that the GSP mechanism is not truthful.

3.5.2 Myerson's Lemma vs GSP

Let's simplify the model a little bit in order to understand why the truthfulness property, which holds in the Second Price auction, doesn't hold anymore. Suppose that $\sigma_i = 1, \forall i$ and $r = 0$. Then, the allocation rule of GSP boils down to just giving the m slots to the m highest bidders respectively. It's easy to see that this allocation rule is monotone, bidding higher can only give you a more valuable slot. Therefore, we know that an immediate consequence of Myerson's Lemma is that there exists a payment rule $\boldsymbol{p'}$ such that the mechanism $(\boldsymbol{x}_{\text{GSP}}, \boldsymbol{p'})$ is truthful.

Myerson's payment rule Let's now see how the payments imposed by Myerson's formula would look like in that setting. W.l.o.g. assume that $b_1 \geq b_2 \geq ... \geq b_n$. Then, for player i we have that

$$c_{i\sigma_i(\boldsymbol{b})}(\boldsymbol{b}) = \sum_{i=1}^{\text{bidders allocated}} \frac{a_j - a_{j+1}}{a_i} b_{j+1}$$

We can see now that Myerson's Lemma suggests that bidder's i payment for each click should be a suitable combination of lower bids. This mechanism is guaranteed to be truthful, and the payment rule described above is the only which when coupled with GSP's allocation rule produces a truthful mechanism. Therefore, we can conclude that the average cost $C_i(\boldsymbol{b})$ for bidder i is

$$C_i(\boldsymbol{b}) = \sum_{i=1}^{\text{bidders allocated}} \gamma_i (a_j - a_{j+1}) b_{j+1} \tag{3.1}$$

Recall that the average cost for bidder i imposed by the GSP mechanism has the following form

$$C_i(\boldsymbol{b}) = a_{\sigma_i(\boldsymbol{b})}\gamma_i b_{\pi_{\sigma_i(\boldsymbol{b})+1}} \tag{3.2}$$

Comparing equations (3.1) and (3.2) we can understanding clearly the reason why the GSP mechanism isn't truthful; it's payment for i rule doesn't involve all the lowest bids, as Myerson's formula suggests, and since Myerson's rule is the only one that can produce a truthful auction we know for sure that GSP isn't truthful.

3.5.3 GSP's properties

Although it is known that GSP isn't truthful it remains the most widely used mechanism for allocating advertising slots in online markets. Therefore it's crucial to understand some of the properties that it possesses. Edelman et al. [EOS07] and Varian [Var07] analyzed some of them. Imagine that the bids of n players reach a stable state, meaning that no player can increase his payoff by deviating to some other bid unilaterally. Assuming again w.l.o.g. that $b_1 \geq b_2 \geq ... \geq b_n$ and that $b_i \geq r, \forall i$, we have that the bid profile $\boldsymbol{b}$ is "stable" if and only if

$$a_i\gamma_i(v_i - b_{i+1}) \geq a_j\gamma_i(v_i - b_{j+1}), j > i \tag{3.3}$$

$$a_i\gamma_i(v_i - b_{i+1}) \geq a_j\gamma_i(v_i - b_j), j < i \tag{3.4}$$

Another interesting property of the GSP mechanism has to do with the revenue that it generates for the auctioneer. Suppose that the bidders reach a different notion of a stable state, called an *envy-free equilibrium*. That is, each bidder i is as happy with his current slot at the price he pays for that as he would be for any other slot in its respective current price. A *locally envy-free equilibrium* is a state where the above condition holds only for each bidder's neighboring slots. In that state, the following theorem holds.

***Theorem* 3.5.1.** *If the number of advertisers is greater than the number of advertising slots then the auctioneer's revenue under any locally envy-free equilibrium of the GSP auction is at least as much as the revenue of the Myerson's derived auction under the assumption that bidders are truthful.*

The above theorem suggests that in order to increase your revenue, if some (mild) assumptions hold, you should use the GSP mechanism instead of the Myerson's payment rule. This is one of the main reasons why it's so widely used. Apart from that, the GSP mechanism protects -in some sense- the bidder's privacy, since the bidder doesn't have to reveal his true value in order to maximize his revenue. Furthermore, the payment rule is much simpler than that of Myerson's derived auction.

Nevertheless there is a key challenge in that setting. The previous analysis assumed that the bidders can reach a stable setting that enjoys some good properties. However, it doesn't give any strategy whatsoever that can help them reach that state. This deep question regarding how bidders should behave in such settings will be analyzed thoroughly in the next chapter.

Chapter 4

Online Learning

In the previous chapter we presented some basic ideas and results regarding Mechanism Design, illustrating some important mechanisms that are used in practice and have strong theoretical guarantees. In this chapter we will switch gears and discuss how players can behave in highly uncertain environments in order to have non trivial performance guarantees. We call these players *learners*, because they are trying to learn from past interactions how they should play in the future. More precisely, we will present a field of Machine Learning which is called *Online Learning* and that has interesting theoretical properties and various practical applications. Online learning is the process of answering a sequence of questions given some (sometimes partial) knowledge of the correct answers to previous questions and possibly other information. This field has elements borrowed from game theory (its measure of "success"), convex optimization and statistical learning theory.

We will start by providing a high level description of the "game" that occurs in online learning. The game proceeds in discrete rounds, and at each time step the learner commits to a decision from a predefined space. After that, the learner suffers a loss which is associated with the decision. Different decisions can produce possibly different losses. These losses aren't known to the decision maker beforehand and aren't necessarily generated stochastically. In fact they can be adversarially chosen or even depend on the action taken by the decision maker at the current time step. We can already see that several restrictions are necessary in order to be able to obtain non-trivial results in that framework.

1. The losses that are determined by the adversary have to be bounded. To see why that is necessary imagine that the adversary keeps decreasing the scale of loss at each time step. In that case, no online algorithm can ever recover from the loss that incurred in the first time step. Therefore, we have to assume that the losses are in some bounded region.

2. The decision set must be somehow bounded and/or structured, otherwise in an infinite decision set the adversary can assign high loss to all strategies chosen by the player indefinitely, while setting some decisions' losses to zero. A situation like that prohibits any meaningful performance benchmark.

4.1 Online Learning Model

Before we define the formal model of online learning is useful to provide two important definitions, regarding *convex sets* and *convex functions*.

Definition 4.1.1. Convex Set
A set $K \subseteq \mathbb{R}^n$ is convex if $\forall \boldsymbol{x}, \boldsymbol{y} \in K, \forall a \in [0,1]$ we have that $a\boldsymbol{x} + (1-a)\boldsymbol{y} \in K$, that is

for every two points in the set, all the points on the line segment connecting them are also in the set.

Definition 4.1.2. Convex Function
A function $f : K \to \mathbb{R}$ is convex if $\forall \boldsymbol{x}, \boldsymbol{y} \in K, \forall a \in [0,1]$ we have that $f(a\boldsymbol{x} + (1-a)\boldsymbol{y}) \leq af(\boldsymbol{x}) + (1-a)f(\boldsymbol{y})$.

We are now ready to define a formal model for Online Learning. Its basic elements are the following.

- a decision set $K \subseteq \mathbb{R}^n$, which is convex
- a bounded set F of cost functions available to the adversary. Every $f_t : K \to \mathbb{R}$ is assumed to be *convex*
- T is the total number of game iterations

The game proceeds in the following way. At each time step t the player commits to a decision $\boldsymbol{x_t} \in K$. After that, the adversary reveals a cost function $f_t \in F$ and the player's loss is $f_t(\boldsymbol{x_t})$, which is the value of the function at his decision point.

What makes an algorithm successful in the setting at hand? Since the framework is game-theoretic the suitable metric for success comes also from game theory and is called *regret*.

Definition 4.1.3. Regret
Let A be an online learning algorithm. We define the regret of A after T rounds as

$$\text{regret}_T(A) = \sup_{\{f_1,\dots,f_T\} \subseteq F} \Big\{ \sum_{t=1}^{T} f_t(\boldsymbol{x_t}) - \min_{\boldsymbol{x} \in K} \sum_{t=1}^{T} f_t(\boldsymbol{x}) \Big\}$$

This metric measures how much the learner regrets not following at each round the best fixed point in hindsight. Intuitively, we want the regret to be a sublinear funtion of T, that is $\text{regret}_T(A) = o(T)$, since then the player will perform on average as well as the best fixed strategy in hindsight. When it is clear from context we will omit the supremum over F.

The running time of an online learning algorithm is defined to be the worst case time needed to produce $\boldsymbol{x_t}$ for an iteration t. Typically, it depends on n, which is the dimension of the decision space, T, the total number of iterations, as well as some other parameters regarding the cost functions and the underlying convex set that we will explore later on.

4.2 Discrete Setting

We will start by exploring algorithms that perform on a discrete decision space. Imagine that you want to answer a sequence of yes/no questions and everytime you guess correctly you suffer a loss of 0, whereas when your answer is wrong you suffer a loss of 1. In addition to the answers of previous rounds, you are given a set of N (presumable) experts who are willing to provide their opinions at the beginning of each round. The problem is that you don't know if these people are actually experts or scammers. What should you do?

4.2.1 Realizable Case

Let's restrict the problem a little bit further. Suppose that at least one of them is actually an expert and will never make a mistake throughout the game. We call that setting the *Realizable Case*. This makes things a lot easier as we only need to identify him. A first approach is the following algorithm, called *Consistent*.

Algorithm 3 Consistent

Input a finite set H of n experts

$V_1 \leftarrow H$
for $t = 1...T$ **do** ▷ We have to answer T questions
    receive question q_t
    choose any expert $h_t \in V_t$
    submit his opinion $p_t = h_t(q_t)$
    receive true answer y_t
    update $V_{t+1} = \{h_t \in V_t | h_t(q_t) = y_t\}$ ▷ Keep the experts who were *consistent* so far

The simple idea is that we keep a set of all the experts who haven't yet made any mistake, because then the expert who makes no mistakes is guaranteed to be among them. Calculating the total number of mistakes that our algorithm makes is also simple. If we make a mistake at step t at least one expert is removed from V_t. Therefore, after making M mistakes we have that $|V_t| \leq |H| - M$ and since we know that $|V_t| \geq 1, \forall t$ we get that $M_{\text{Consistent}}(H) \leq |H| - 1$. Since, in the worst case, we remove experts linearly from our initial set, we get as expected a linear bound on the number of mistakes that the algorithm makes. However, this is not satisfactory since H can be very large in many cases of interest. Can we improve that?

The answer is positive. The problem with the approach described above is that (in the worst case) after each mistake we evict only one expert from our consistent set. We will modify our strategy in order to obtain a logarithmic bound on the number of mistakes. This new algorithm is called *Halving*.

Algorithm 4 Halving

Input a finite set H of n experts

$V_1 \leftarrow H$
for $t = 1...T$ **do** ▷ We have to answer T questions
    receive question q_t
    predict $p_t = arg\max_{r \in \{0,1\}} |\{h_t \in V_t : h_t(q_t) = r\}|$ ▷ Break ties arbitrarily
    submit p_t
    receive true answer y_t
    update $V_{t+1} = \{h_t \in V_t | h_t(q_t) = y_t\}$ ▷ Same idea as before

What we do now is that instead of picking a single expert from the consistent set, we consult every one of them and follow the opinion of the majority. In that way, whenever our prediction is wrong we don't reduce our set by one expert, we *halve* it because at least half of the experts in our consistent set were wrong in that turn. Therefore, we get that $M_{\text{Halving}}(M) \leq \log_2(|H|)$, since $1 \leq |V_{t+1}| \leq |H|2^{-M}$. We can see that by modifying our initial, seemingly naive, approach we get an exponential improvement in the number of mistakes that our algorithm makes. Next we will use the idea of following the majority a little differently, in order to obtain regret bounds for the general problem in the discrete setting.

4.2.2 Multiplicative Weights Update

Let's now consider the case where each expert i has his own opinion and after each round t is associated with a loss, $l_t(i)$, which is a non-negative number. The losses are not binary as before and we do not know that there exists an expert who is never wrong. The algorithm that we present below guarantees that the average expected loss of the player is approaching that of the best expert in hindsight. The idea is to assign weights to each expert and to penalize each one according to his loss in the previous round, in an exponential fashion, so that very quickly the whole "mass" will be concentrated on the best experts. This algorithm is called *Multiplicative Weights Update* and has been rediscovered independently in many different fields (in game theory see e.g. [Bro51, BVN50, Rob51], in machine learning [Lit88, LW94]). For an extensive presentation of the various applications of that algorithm the interested reader is referred to [AHK12].

Algorithm 5 Multiplicative Weights Update

Initialize: $\forall i \in [N], W_1(i) = 1$
for $t = 1...T$ **do** ▷ We have to answer T questions
  receive question q_t
  pick i_t according to W_t, i.e. $\mathbb{P}[i_t = i] = \boldsymbol{x}_t(i) = \frac{W_t(i)}{\sum_{j=1}^{n} W_t(j)}$
  suffer loss $l_t(i_t)$
  update weights $W_{t+1}(i) = W_t(i)e^{-\epsilon l_t(i)}, \forall i$

We can see that the expected loss of that algorithm at round t is $\mathbb{E}[l_t(i_t)] = \sum_{i=1}^{N} \boldsymbol{x}_t(i) l_t(i) = \boldsymbol{x}_t^\top l_t$, therefore, on expectation, the total expected loss of MWU is $\sum_{t=1}^{T} \boldsymbol{x}_t^\top l_t$. The following theorem provides a bound on the regret of the algorithm.

***Theorem* 4.2.1.** *Let l_t^2 denote the n-dimensional vector of square losses, i.e. $l_t^2(i) = l_t(i)^2$, let $\epsilon > 0$ and assume all losses to be non-negative. Then, for any expert $i^* \in [n]$ the algorithm satisfies*

$$\sum_{t=1}^{T} \boldsymbol{x}_t^\top l_t \leq \sum_{t=1}^{T} l_t(i) + \epsilon \sum_{t=1}^{T} \boldsymbol{x}_t^\top l_t^2 + \frac{\log n}{\epsilon}$$

Proof. Let $\Phi_t = \sum_{i=1}^{n} W_t(i)$ be the sum of the weights assigned to the experts. After round t we have that

$$\begin{aligned}
\Phi_{t+1} &= \sum_{i=1}^{n} W_t(i) e^{\epsilon l_t(i)} \\
&= \Phi_t \sum_{i=1}^{n} \boldsymbol{x}_t(i) e^{\epsilon l_t(i)} \\
&\leq \Phi_t \sum_{i=1}^{n} \boldsymbol{x}_t(i)(1 - \epsilon l_t(i) +) \epsilon^2 l_t^2(i)) \\
&= \Phi_t (1 - \epsilon \boldsymbol{x}_t^\top l_t + \epsilon^2 \boldsymbol{x}_t^\top l_t^2) \\
&\leq \Phi_t e^{-\epsilon \boldsymbol{x}_t^\top l_t + \epsilon^2 \boldsymbol{x}_t^\top l_t^2}
\end{aligned}$$

Observe that $\Phi_0 = N$, therefore $\Phi_T \leq N e^{-\epsilon \sum_{t=1}^{T} \boldsymbol{x}_t^\top l_t + \epsilon^2 \sum_{t=1}^{T} \boldsymbol{x}_t^\top l_t^2}$. On the other hand, by definition, for i^* we get that

$$W_T(i^*) = e^{-\epsilon \sum_{t=1}^{T} l_t(i^*)}$$

We know that $W_T(i^*) \leq \Phi_T$, thus

$$W_T(i^*) \leq N e^{-\epsilon \sum_{t=1}^{T} \boldsymbol{x}_t^\top l_t + \epsilon^2 \sum_{t=1}^{T} \boldsymbol{x}_t^\top l_t^2}$$

The theorem follows by taking the logarithm of both sides and simplifying. □

4.3 Continuous Setting

We will now switch gears and discuss the continuous model, which was defined in the previous sections. Before doing that, it is worth analyzing the so-called *offline* case which serves as a motivation for one of the most important algorithms that we will present.

4.3.1 Offline Gradient Descent

In the offline convex optimization setting we are given a convex function f defined over a convex domain K and we are asked to find its (global) minimum. Since the function is convex, we know that a local optimum translates to a global optimum, so our task reduces to simply finding a local minimum of the function. Let's define first some notions which will be useful.

- diameter D of a set K is a value such that $||\boldsymbol{x} - \boldsymbol{y}|| \leq D, \forall \boldsymbol{x}, \boldsymbol{y} \in K$, where $||\cdot||$ is some norm defined on K
- if $f : K \to \mathbb{R}$ is differentiable, then it is convex if and only if $\forall \boldsymbol{x}, \boldsymbol{y} \in K$ we have

  $$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$$

  If f is convex but non-differentiable, then the subgradient of f at $\boldsymbol{x}$ is defined to be any member of the set $\{\nabla f(\boldsymbol{x})\}$ that satisfies the above inequality for all $\boldsymbol{y}$.
- a function f is Lipschitz continuous with parameter G if $\forall \boldsymbol{x}, \boldsymbol{y} \in K$

  $$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq G||\boldsymbol{x} - \boldsymbol{y}||$$

  If the norm of the subgradient of f is bounded by G then f is Lipschitz continuous with parameter G.
- f is α-strongly convex if

  $$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\alpha}{2}||\boldsymbol{y} - \boldsymbol{x}||^2$$

- f is β-smooth if

  $$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta}{2}||\boldsymbol{y} - \boldsymbol{x}||^2$$

  This condition is equivalent to a Lipschitz condition over the gradients, with parameter β, i.e.

  $$||\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})|| \leq \beta||\boldsymbol{x} - \boldsymbol{y}||$$

- if a function is α-strongly convex and β-smooth then it is called γ-well-conditioned, where $\gamma = \frac{\alpha}{\beta}$, and it holds that $\gamma \leq 1$.
- a function f is is δ-exp-concave over K if the function g is concave, where $g : K \to \mathbb{R}$ is

  $$g(\boldsymbol{x}) = e^{-af(\boldsymbol{x})}$$

- a projection of a point $\boldsymbol{y} \in R^n$ to a set K with respect to norm $||\cdot||$, is defined as $\Pi_K(\boldsymbol{y}) = arg\min_{\boldsymbol{x} \in K} ||\boldsymbol{y} - \boldsymbol{x}||$.

Algorithm 6 Gradient Descent

1: Initialize: pick $\boldsymbol{x}_0 \in K$
2: **for** $t = 1...T$ **do** ▷ We make T iterations
3: $\quad \boldsymbol{y}_t = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} = \Pi_K(\boldsymbol{y}_{t+1})$ ▷ Project to K to maintain feasibility
4: **return** $\boldsymbol{x}_{T+1}$

A very intuitive and straightforward algorithm that achieves that goal is *gradient descent*, an iterative algorithm which starts from an arbitrary point of the set and at each step moves to the direction of the steepest descent of the function. By making careful steps towards that direction it ends up abitrarily close to the optimum. The convergence rate of the above algorithm is given by the following theorem.

***Theorem* 4.3.1.** *For step size* $\eta = \frac{D}{G\sqrt{T}}$ *the sequence* $\boldsymbol{x}_1, ..., \boldsymbol{x}_T$ *of the above algorithm satisfies*

$$f(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t) \leq \min_{\boldsymbol{x}^* \in K} f(\boldsymbol{x}^*) + \frac{DG}{\sqrt{T}}$$

where D *is the diameter of the set* K *and* G *is an upper bound on the norm of the gradients.*

Proof. Start by observing that

$$||\boldsymbol{x}^* - \boldsymbol{y}_{t+1}||^2 = ||\boldsymbol{x}^* - \boldsymbol{x}_t||^2 - 2\eta \nabla f(\boldsymbol{x}_t)(\boldsymbol{x}_t - \boldsymbol{x}^*) + \eta^2 ||\nabla f(\boldsymbol{x}_t)||^2$$

Moreover, Pythagoras' theorem implies

$$||\boldsymbol{x}^* - \boldsymbol{x}_{t+1}|| \leq ||\boldsymbol{x}^* - \boldsymbol{y}_{t+1}||$$

Thus, we have that

$$||\boldsymbol{x}^* - \boldsymbol{x}_{t+1}|| \leq ||\boldsymbol{x}^* - \boldsymbol{x}_t||^2 - 2\eta \nabla f(\boldsymbol{x}_t)(\boldsymbol{x}_t - \boldsymbol{x}^*) + \eta^2 G^2 \tag{4.1}$$

We obtain the result by exploiting the convexity of f.

$$\begin{aligned}
&f(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t) - f(\boldsymbol{x}^*) \\
&\leq \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \nabla f(\boldsymbol{x}_t)(\boldsymbol{x}_t - \boldsymbol{x}^*) \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \frac{1}{2\eta}(||\boldsymbol{x}^* - \boldsymbol{x}_{t+1}||^2 - ||\boldsymbol{x}^* - \boldsymbol{x}_t||^2) + \frac{\eta}{2}G^2 \\
&\leq \frac{1}{2\eta T}D^2 + \frac{\eta}{2}G^2 \leq \frac{DG}{\sqrt{T}}
\end{aligned}$$

The first two inequalities follow from the convexity of f, the next from 4.1 and the last from summing the telescoping series.

□

An immediate corollary of the previous theorem is that in order to get an ϵ-approximate solution one needs to apply $O(\frac{1}{\epsilon^2})$ gradient iterations.

4.3.2 Online Gradient Descent

We are now ready to tackle the original problem that we were interested in. We briefly remind the setting; at each time step t the learner picks a point $\boldsymbol{x}_t \in K$ from a convex domain K, then the adversary picks a convex loss function f_t and the learners suffer a loss $f_t(\boldsymbol{x}_t)$. The goal for the learner is to be competitive with the best fixed point in hindsight, i.e. to incur on average the same loss as that point. As it turns out, by carefully tuning the step size η_t, we can use the exact algorithm we described above in order to achieve vanishing regret. The online version of gradient descent and the general *online convex optimization*(OCO) framework is due to Zinkevich [Zin03].

Algorithm 7 Online Gradient Descent

1: **for** $t = 1...T$ **do** ▷ We make T iterations
2: play $\boldsymbol{x}_t$ and observe loss $f_t(\boldsymbol{x}_t)$
3: $\boldsymbol{y}_t = \boldsymbol{x}_t - \eta_t \nabla f_t(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} = \Pi_K(\boldsymbol{y}_{t+1})$ ▷ Project to K to maintain feasibility
4: **return** $\boldsymbol{x}_{T+1}$

***Theorem* 4.3.2.** *Online gradient descent with step size $\eta_t = \frac{D}{G\sqrt{t}}$ gurantees the following regret bound:*

$$regret_T = \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{x^* \in K} \sum_{t=1}^{T} f_t(\boldsymbol{x}^*) \leq \frac{3}{2} GD\sqrt{T}$$

The proof of the theorem is very similar to the one in the offline case, as we use an inequality similar to 4.1 and exploit the convexity of the loss functions. At first glance it might not make sense to move in the direction of the gradient of the previous loss function, after all the function in the next round might be very different than the previous one. In order to make it seem more intuitive we can view the whole process a little differently; imagine that we want to optimize the function $F(\boldsymbol{x}) = \sum_{t=1}^{T} f_t(\boldsymbol{x})$ that we do not know beforehand and at each time step we are given one new "component". Since gradient is a linear operator, in order to move to the direction of the gradient of function $F(\boldsymbol{x})$ we can simply move to the direction of the gradients of functions $f_t(\boldsymbol{x})$ and add those steps.

We note that in the previous algorithm we assume that the learner has an oracle access to the gradients of the loss functions and we do not care about the computational cost of evaluating neither the gradient nor the projection to K. It is worth mentioning that this setting is called *full information*, since the learner has access to the losses of all points in K. In many interesting applications this is not true, as the learner can only find out the loss that he suffers. In that case we have the so-called *bandit* setting. For an excellent survey of that topic by Bubeck and Cesa-Bianchi the interested reader is referred to [BCB+12].

4.3.3 Follow the Leader

In this section we will describe a (meta)algorithm that guarantees sublinear regret, which is more intuitive than the one discussed above. The most natural thing to do at round t would be to simply pick the point with the minimum loss in the first $t-1$ rounds and hope that it continues its success. That's the reason why this algorithm is called *Follow the Leader*(FTL). Formally, we have that

$$\boldsymbol{x}_{t+1} = arg\min_{\boldsymbol{z} \in K} \sum_{t=1}^{T} f_t(\boldsymbol{z}), \forall t$$

A useful property of that strategy is that the cumulative regret is bounded by the cumulative difference between the loss of $\boldsymbol{x}_t, \boldsymbol{x}_{t+1}$.

***Lemma* 4.3.3.** *Let $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T$ be the sequence of points produced by FTL. Then, $\forall x^* \in K$ it holds that*

$$\textit{regret}_T(\boldsymbol{x}^*) = \sum_{t=1}^{T}(f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}^*)) \leq \sum_{t=1}^{T}(f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_{t+1}))$$

Proof. Subtracting $\sum_{t=1}^{T}(f_t(\boldsymbol{x}_t)$ from both sides we need to show, equivalently, that

$$\sum_{t=1}^{T} f_t(\boldsymbol{x}_{t+1}) \leq \sum_{t=1}^{T} f_t(\boldsymbol{x}^*)$$

We will prove it using induction. The base case of $T = 1$ follows directly from the definition of $\boldsymbol{x}_{t+1}$. Assume that the inequality holds for $T - 1$, then for all $\boldsymbol{x}^* \in K$ we have

$$\sum_{t=1}^{T-1} f_t(\boldsymbol{x}_{t+1}) \leq \sum_{t=1}^{T-1} f_t(\boldsymbol{x}^*)$$

Adding $f_t(\boldsymbol{x}_{t+1})$ to both sides we get

$$\sum_{t=1}^{T} f_t(\boldsymbol{x}_{t+1}) \leq \sum_{t=1}^{T-1} f_t(\boldsymbol{x}^*) + f_t(\boldsymbol{x}_{+1})$$

Since the above holds $\forall \boldsymbol{x}^* \in K$, it holds for $\boldsymbol{x}^* = \boldsymbol{x}_{+1}$, therefore we get

$$\sum_{t=1}^{T} f_t(\boldsymbol{x}_{t+1}) \leq \sum_{t=1}^{T} f_t(\boldsymbol{x}_{T+1}) = \min_{\boldsymbol{x}^* \in K} \sum_{t=1}^{T} f_t(\boldsymbol{x}^*)$$

□

Since f_t are Lipschitz continuous functions, the above theorem shows us that if the predictions of FTL, i.e. the best points so far, are relatively stable then the regret of the algorithm will be low. Therefore, we can see that stability plays a very significant role in the performance of this strategy. For instance, if the loss functions are quadratic, namely $f_t(\boldsymbol{x}) = \frac{1}{2}||\boldsymbol{x} - \boldsymbol{z}_t||^2$, then FTL guarantees regret of at most $4L^2(\log(T) + 1)$, where $L = \max_t ||\boldsymbol{z}_t||$. The reason for that is that the predictions vary little as time goes by.

Unfortunately, this is not the case for linear loss functions. Consider the following counter example. Let $K = [-1, 1]$ and $f_t(\boldsymbol{x}) = \boldsymbol{x} \cdot \boldsymbol{z_t}$ where

$$\boldsymbol{z}_t = \begin{cases} -0.5 & 0 \leq t = 1 \\ 1 & t \bmod 2 = 0 \\ -1 & t > 1 \wedge t \bmod 2 = 1 \end{cases}$$

In that case, FTL will predict $\boldsymbol{x}_t = 1$ when t is odd and $\boldsymbol{x}_t = -1$ when t is even. Thus, its total loss will be T, while the cumulative loss of the best point in hindsight is 0 ($\boldsymbol{x}^* = \boldsymbol{0}$). The reason why the algorithm fails in that case is that the predictions are very unstable, adding one more function vastly changes the point that it answers.

In order to fix that problem we will tweek the algorithm a little bit by adding a *regularizer*, which is a function $R : K \to \mathbb{R}$, that will ensure the stability of the decisions. Intuitively, the

regularizer prevents a situation similar to overfitting, that is choosing a point that minimizes the loss in the previous rounds but is not able to "generalize" well in the next round. By not minimizing exactly that loss, the algorithm is able to make relatively stable decisions, which in turn ensure low regret. This version of Follow the Leader is referred to as Follow the *Regularized* Leader(RFTL). Formally, we now have

$$\boldsymbol{x}_{t+1} = arg\min_{\boldsymbol{z}\in K} \sum_{t=1}^{T} f_t(\boldsymbol{z}) + R(\boldsymbol{z}), \forall t$$

Naturally, different choices of regularization functions will yield different algorithms, that is why RFTL is sometimes called a *meta* algorithm. Before we provide the regret bounds of this modified approach, observe that since f_t is convex, the regret of every sequence $f_1, ..., f_T$ can be upper bounded by $\sum \nabla f_t(\boldsymbol{x}_t)(\boldsymbol{x}_t - \boldsymbol{x}^*)$. Therefore, it suffices to provide regret bounds against a sequence of linear functions.

***Theorem* 4.3.4.** *Consider running FTRL against a sequence of linear functions, i.e.* $f_t(\boldsymbol{x}) = \boldsymbol{x} \cdot \boldsymbol{z}_t$*, with* $K = \mathbb{R}^n$*, and with the regularizer* $R(\boldsymbol{x}) = \frac{1}{2\eta}||\boldsymbol{x}||_2^2$*. Then* $\forall \boldsymbol{x}^* \in K$ *we have*

$$regret_T \leq \frac{1}{2\eta}||\boldsymbol{x}^*||^2 + \eta \sum_{t=1}^{T} ||\boldsymbol{z}_t||_2^2$$

Moreover, if we assume that $||\boldsymbol{x}^*|| \leq B, \forall \boldsymbol{x}^* \in K$ *and* $\frac{1}{T}\sum_{t=1}^{T} ||\boldsymbol{z}_t||_2^2 \leq L^2$*, then setting* $\eta = \frac{B}{L\sqrt{2T}}$ *we get*

$$regret_T \leq BL\sqrt{2T}$$

Before proving that theorem we have to figure out how will the decision of the algorithm look like. We can easily verify that $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{z}_t$, and since $\boldsymbol{z}_t$ is the derivative of the loss function, RFTL with that regularizer yields Online Gradient Descent. Using that formula for $\boldsymbol{x}_t$ we can prove the previous theorem by simply bounding the stability of the decision points.

4.4 Lower Bounds on Regret

So far we have seem some algorithms that achieve sublinear cumulative regret in discrete as well as continuous settings. A natural question to ask is how low can the regret of any online learning algorithm be? The following theorem provides an answer to that qestion.

***Theorem* 4.4.1.** *The regret of any online learning algorithm is* $\Omega(DG\sqrt{T})$ *in the worst case, where* D *is the diameter of the decision set and* G *is a bound on the norm of the gradients. This holds even when the losses are generated from a fixed stationary distribution.*

We will provide a sketch of the proof. Consider an instance of online learning where $K = \{\boldsymbol{x} \in \mathbb{R}^n, ||\boldsymbol{x}||_\infty \leq 1\}$. There are 2^n linear cost functions, one for each vertex $\boldsymbol{v} \in \{\pm 1\}^n$ defined as

$$\forall \boldsymbol{v} \in \{\pm 1\}^n, f_{\boldsymbol{v}}(\boldsymbol{x}) = \boldsymbol{v} \cdot \boldsymbol{x}$$

Notice that for the diameter of K and for the gradients of the functions it holds that

$$D \leq \sqrt{\sum_{i=1}^{n} 2^2} = 2\sqrt{n}, G \leq \sqrt{\sum_{i=1}^{n} (\pm 1)^2} = \sqrt{n}$$

At each time step one of these 2^n functions is chosen uniformly and independently at random, thus for any t and any $\boldsymbol{x}_t \in K$ we have that $\mathbb{E}_{\boldsymbol{v}_t}[f_t(\boldsymbol{x}_t)] = \mathbb{E}_{\boldsymbol{v}_t}[\boldsymbol{v}_t \cdot \boldsymbol{x}_t] = 0$.

However $\mathbb{E}_{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_T}[\min_{\boldsymbol{x}\in K}\sum_{t=1}^{T} f_t(\boldsymbol{x})] = \mathbb{E}_{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_T}[\min_{\boldsymbol{x}\in K}\sum_{i=1}^{n}\sum_{t=1}^{T} \boldsymbol{v}_t(i)\boldsymbol{x}_t] = n\mathbb{E}_{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_T}[-|\sum_{t=1}^{T} \boldsymbol{v}_t(1)|] = -\Omega(n\sqrt{T})$, where we get the second to last equality because the coordinates are i.i.d.. The last one follows by bounding the difference between the heads and tails in a sequence of T (fair) coin tosses.

Until now, we have discussed some algorithms that guarantee $O(\sqrt{T})$ regret in the general setting, and we proved that (in the worst case) we cannot do any better than that. So is that all the online learning framework has to offer? The answer is negative. As it is many times the case, by making some more assumptions on the loss functions we can design algorithms that guarantee improved regret bounds. The following table summarizes them.

	a-strongly convex	β-smooth	δ-exp-concave
Upper Bound	$\frac{1}{a}\log T$	$\sqrt{T}$	$\frac{n}{\delta}\log T$
Lower Bound	$\frac{1}{a}\log T$	$\sqrt{T}$	$\frac{n}{\delta}\log T$
Average Regret	$\frac{\log T}{aT}$	$\frac{1}{\sqrt{T}}$	$\frac{n\log T}{\delta T}$

It is worth mentioning that in the offline case, smoothness improves the convergance rate to the optimum, whereas (as we see above) in the online it does not. On the other hand, exp-concavity does not improve offline covergence rate, but it helps in the online setting.

4.5 More Notions of Regret

Thus far we have studied algorithms that minimize *static* regret, in the sense that the optimum point that they compete against , although unknown beforehand, is fixed. Using that metric makes sense in many cases, for instance when there is an underlying distribution generating the losses these learning algorithms manage to "learn" that distribution and approach the optimal strategy. However, there are instances in which it does not make sense to compete against a fixed point. Imagine that the underlying distribution that generates the losses varies slowly over time. Then, the algorithms that we presented before will converge to an "average" solution that is not instead of adapting to the changing environment. We will present two more notions of regret which serve as benchmarks for algorithms that are constantly trying to adapt to those changes.

4.5.1 Adaptive Regret

Hazan and Seshadhri [HS07] introduced the notion of *adaptive regret* to measure the perfomance of algorithms that are learning to play in a contiously changing environment, and provided an efficient scheme to convert any low (static) regret algorithm to a low adaptive regret algorithm. Intuitively, this metric requires that the learning algorithm achieves low regret in any sufficiently large continuous interval. Formally, we have

$$\text{Adaptive-Regret}(A) = \sup_{I=[r,s]\subset[T]} \{\sum_{t=r}^{s} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x}^*\in K}\sum_{t=r}^{s} f_t(\boldsymbol{x}^*)\} \tag{4.2}$$

It follows directly from the definition that adaptive regret strictly generalizes static regret. Also, notice that if a learning algorithm A guarantees $O(R)$ adaptive regret then it converges to a "local" optimum in each interval of length $\Omega(R)$.

The reason why algorithms that minimize regret fail under that new metric is that they treat equally all the past iterations, while in fact it is often the case that the distant past should not affect the decisions very much. For instance, consider using FTL in a game that

$K = [-1, 1]$, while the adversary chooses $f_t(x) = (x-1)^2$ for the first $T/2$ iterations and $f_t(x) = (x+1)^2$ in the last $T/2$. The algorithm will set $x_t = 1$ in the first $T/2$ while slowly moving towards 0 in the last $T/2$. Although static regret is $O(\log T)$ we notice that under this new benchmark this algorithm has regret $\Omega(T)$, since in the last $T/2$ iterations the best fixed point is -1.

The solution to overcome that problem is to design algorithms that are biased towards more recent outcomes of the learning process, so that the distant past will not affect the outcome that much. The two methods, *Follow the Leading History*(FLH) and *Advanced Follow the Leading History*(AFLH), that Hazan and Seshadhri proposed guarantee the following bounds.

***Theorem* 4.5.1.** *Suppose the functions $f_1, ..., f_T$ are convex and bounded by M and there exists an algorithm giving $R(T)$ regret with running time V. The running time of algorithm FLH is $O(VT)$ and Adaptive-Regret(FLH) $\leq R(T) + 0(M\sqrt{T \log T})$. The running time of algorithm AFLH is $O(V \log T)$ and Adaptive-Regret(AFLH) $\leq R(T) \log T + 0(M\sqrt{T \log^3 T})$*

4.5.2 Dynamic Regret

Another benchmark that has been proposed to measure the performance of learning algorithms which try to operate in a changing environment is the *dynamic regret* (see e.g. [BGZ15, HW15, JRSS15, MSJR16]). This metric, instead of comparing the total loss of the algorithm to that of the best fixed point, it compares it with a sequence of changing points that possibly correspond to locally optimal solutions. If this sequence varies "slowly" these learning algorithms that can adapt to those changes. Formally, the dynamic regret is defined as

$$\text{regret}_T^d(\boldsymbol{x}_1^*, ..., \boldsymbol{x}_T^*) = \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{x}_t^*) \tag{4.3}$$

Usually, the variation of the sequence is quantified using the follow 4 metrics.

$$V_T = \sum_{t=2}^{T} \sup_{\boldsymbol{x} \in K} |f_t(\boldsymbol{x}) - f_{t-1}(\boldsymbol{x})| \tag{4.4}$$

This quantity measures how different are the consecutive loss functions that are chosen from the adversary, using the supremum of their difference over all points of the decision set.

$$D_T = \sum_{t=1}^{T} ||\nabla f_t(\boldsymbol{x}_t) - M_t||^2 \tag{4.5}$$

where M_t is a prediction that is available to the algorithm at time t (for instance, it could be $\nabla f_t(\boldsymbol{x}_t)$). This metric measures how much the gradients of the loss functions vary over time.

$$C_T(\boldsymbol{u}_1, ..., \boldsymbol{u}_T) = \sum_{t=2}^{T} ||\boldsymbol{u}_t - \boldsymbol{u}_{t-1}|| \tag{4.6}$$

$$C'_T(\boldsymbol{u}_1, ..., \boldsymbol{u}_T) = \sum_{t=2}^{T} ||\boldsymbol{u}_t - \Phi_t(\boldsymbol{u}_{t-1})|| \tag{4.7}$$

where $\Phi_t(\boldsymbol{u}_{t-1})$ is the predicted point by the learner at time t, using information from the previous round. These two metrics quantify the variation of the sequence of points that the learning algorithm is competing against.

The following table summarizes the dynamic regret bounds using the variation metrics that were defined above.

Regret notion	Loss function	Regret rate
$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t)$	Convex	$\mathcal{O}\left(\sqrt{T}(1 + C_T(\mathbf{u}_1, \ldots, \mathbf{u}_T))\right)$
$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{u}_t)$	Convex	$\mathcal{O}\left(\sqrt{T}(1 + C'_T(\mathbf{u}_1, \ldots, \mathbf{u}_T))\right)$
$\sum_{t=1}^{T} \mathbb{E}\left[f_t(\mathbf{x}_t)\right] - f_t(\mathbf{x}_t^*)$	Convex	$\mathcal{O}\left(T^{2/3}(1 + V_T)^{1/3}\right)$
$\sum_{t=1}^{T} \mathbb{E}\left[f_t(\mathbf{x}_t)\right] - f_t(\mathbf{x}_t^*)$	Strongly convex	$\mathcal{O}\left(\sqrt{T(1 + V_T)}\right)$
$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)$	Convex	$\mathcal{O}\left(\sqrt{D_T + 1} + \min\left\{\sqrt{(D_T + 1)C_T}, [(D_T + 1)V_T T]^{1/3}\right\}\right)$
$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)$	Strongly convex	$\mathcal{O}\left(1 + C_T\right)$

Figure 4.1: Dynamic Regret Bounds

Chapter 5

Analyzing Markets under Learning assumptions

The classical approach to analyze strategic interactions between agents was to assume that the players somehow manage to reach a stable state, i.e. a *Nash equilibrium*. This was also the standard econometric approach when trying to infer various information from observed data (see e.g. [AN10, BHN13, JLB07]). The idea behind this approach is straightforward; the distribution of players' actions is observed in the data and if we assume that each player best responds to that distribution we can recover this response from the data. Then, we can invert each player's best response function in order to obtain the private parameter that we are interested in. However, there are several drawbacks regarding this approach. To start with, Daskalakis et al. [DGP09] showed that computing a Nash equilibrium in the general case is a hard problem, therefore it is unrealistic to expect that the outcomes of such games will eventually reach that state. Moreover, even if we assume that somehow players can reach a Nash equilibrium, since it is many times the case that more than just one them exist the "inversion" of the function that we mentioned above can be computationally hard.

Recently, learning outcomes have emerged as attractive alternatives to Nash equilibria. This solution concept is very well suited for online environments, such as internet advertising auctions, which can be thought of as strategic interactions in a constantly changing enviroment. In that setting players have to constantly update their strategies in order to adapt to those variations. For that reason, there have been developed many sophisticated bidding tools that are used by high volume advertisers. Nekipelov et al. [NST15] initiated a line of work which explores properties of such stratetigic interactions under *learning assumptions*, meaning that players use some type of no regret algorithms, like the ones we discussed in the previous section. They explored the problem of inferring a bidder's private valuation using past data in sponsored search auctions, under the assumption that the bidder achieves vanishing (average) regret over time. Subsequently, several works explored the *efficiency* (welfare) guarantees of games in which players use these learning algorithms (see e.g. [SALS15, LST16, FLL+16]). More recently, Braverman et al. [BMSW18] proposed selling strategies for the auctioneer in the single item-single buyer setting, where the bidder is a learning agent, that maximize the seller's revenue. In the following sections we will discuss some of these results further.

5.1 Inferring Valuations in Sponsored Search Auctions

We will start by presenting the valuation inference method proposed in [NST15]. We briefly remind the setting of the sponsored search auctions; at each round, there is a set of n bidders that participate in the auction, each bidder i submits a bid b_i, and the mechanism decides which slot will be allocated to each bidder, if any. We assume that each bidder's utility at round t is $u_{it}(\boldsymbol{b}, v_i) = v_i p_{it}(\boldsymbol{b}) - c_{it}(\boldsymbol{b})$, where $p_{it}(), c_{it}()$ are the click probability and payment functions which are determined by the underlying mechanism (e.g. GSP). Notice that the valuation of each bidder remains constant. The assumption that bidders are learning

agents implies that

$$\frac{1}{T}\sum_{t=1}^{T} u_i(\boldsymbol{b}, v_i) \geq \frac{1}{T}\sum_{t=1}^{T} u_i((b', \boldsymbol{b}_{-i}), v_i) - \epsilon_i, \forall b' \in B_i \tag{5.1}$$

Using the definition of $u_i()$ the above inequality translates to

$$v_i \frac{1}{T}\sum_{t=1}^{T}(p_{it}(b', \boldsymbol{b}_{-i}) - p_{it}(\boldsymbol{b})) \leq \frac{1}{T}\sum_{t=1}^{T}(c_{it}(b', \boldsymbol{b}_{-i}) - c_{it}(\boldsymbol{b})) - \epsilon_i, \forall b' \in B_i \tag{5.2}$$

Since we are focused on inferring the valuation of player i from now on we will drop the dependence on i. Moreover, we denote

$$\Delta P(b') = \frac{1}{T}\sum_{t=1}^{T}(p_{it}(b', \boldsymbol{b}_{-i}) - p_{it}(\boldsymbol{b})) \tag{5.3}$$

the increase in the average click probability when bidder i switches to bid b' and with

$$\Delta C(b') = \frac{1}{T}\sum_{t=1}^{T}(c_{it}(b', \boldsymbol{b}_{-i}) - c_{it}(\boldsymbol{b})) \tag{5.4}$$

the increase in the cost per click from that move. Therefore, we have that no regret learning implies that

$$v\Delta P(b') \leq \Delta C(b') + \epsilon, \forall b' \in B \tag{5.5}$$

We define the *Rationalizable Set*(NR) to be the set of all (v, ϵ) pairs that satisfy the above inequality. As it turns out, this set has some useful properties that make its estimation from data a computationally efficient task.

***Lemma* 5.1.1.** *The rationalizable set is closed and convex.*

The proof of the above lemma follows by the fact that NR is defined by a set of linear inequalities, hence it is convex. It is also closed since the points that satisfy the constraints with equality are also included in the set.

Another useful property is that for each error level ϵ we can easily find all the valuations v such that the (v, ϵ) pairs are rationalizable. Notice that the larger ϵ gets, the more pairs are rationalizable under that value. The following lemma provides the characterization of that pairs.

***Lemma* 5.1.2.** *For any error level ϵ, the valuations that belong in the rationalizable set are in the interval*

$$v \in \Big[\max_{b':\Delta P(b')<0} \frac{\Delta C(b') + \epsilon}{\Delta P(b')}, \min_{b':\Delta P(b')>0} \frac{\Delta C(b') + \epsilon}{\Delta P(b')}\Big]$$

Proof. NR is the set that contains all (v, ϵ) pairs such that $v\Delta P(b) \leq \Delta C(b) + \epsilon, \forall b \in B$. Therefore, if $\Delta P(b) < 0$ we get that $v \geq \frac{\Delta C(b)+\epsilon}{\Delta P(b)} \geq \max_{b':\Delta P(b')<0} \frac{\Delta C(b')+\epsilon}{\Delta P(b')}$.
Similarly, if $\Delta P(b) > 0$ we get that $v \leq \frac{\Delta C(b)+\epsilon}{\Delta P(b)} \leq \min_{b':\Delta P(b')>0} \frac{\Delta C(b')+\epsilon}{\Delta P(b')}$. □

It is worth mentioning that this lemma provides a way to infer the rationalizable set from data. Since $\Delta P(), \Delta C()$ are observable functions, if we discretize the error space and the bid space we can easily calculate the quantities that come into play in the previous lemma.

The problem with the aforementioned approach is that, due to the very high volume of the data that are generated in sponsored search auctions, it might be very inefficient to use all of them when we try to infer the rationalizable set. Therefore, one natural approach is to use only a sample of that data in order to approximate the set that we are interested in. As it turns out, we can leverage the convexity of the set and use its support function representation in order to derive good approximation guarantees, under mild assumptions about the click probability and cost functions. Before we state the theorem regarding the approximation error we need to define some terms.

Definition 5.1.1. The Hausdorff distance $d_H(A, B)$ for subsets A, B of the metric space E with metric $\rho(\cdot, \cdot)$ is defined as

$$d_H(A, B) = \max\{\sup_{a \in A} \inf_{b \in B} \rho(a, b), \sup_{b \in B} \inf_{a \in A} \rho(a, b)\}$$

The Hausdorff distance is used to measure how far two subsets A, B of a metric space are from each other.

Definition 5.1.2. The support function of a closed convex set X is defined as

$$h(X, \boldsymbol{u}) = \sup_{\boldsymbol{x} \in X} \boldsymbol{x} \cdot \boldsymbol{u}$$

The support function representation of a convex set X is very convenient because, as it turns out $d_H(A, B) = \sup_{\boldsymbol{u}} |h(A, \boldsymbol{u}) - h(B, \boldsymbol{u})|$, thus in order to approximate the rationalizable set we simply need to find a good estimator of its support function. If we define $f(\cdot) = \Delta C(\Delta P^{-1}(\cdot))$ using the support function of the rationalizable set, we can prove that $d_H(NR, \widehat{NR}) \leq \sup_z |f(z) - \hat{f}(z)|$, where $\widehat{NR}, \hat{f}(\cdot)$ are the approximations we get via subsampling. We state the formal theorem for the estimation of the set NR below.

***Theorem* 5.1.3.** *Suppose that function f has derivative up to order $k \geq 0$ and for some $L \geq 0$*

$$|f^{(k)}(z_1) - f^{(k)}(z_2)| \leq L|z_1 - z_2|^a$$

Then, we have

$$d_H(NR, \widehat{NR}) \leq O((N^{-1} \log N)^{\frac{\gamma}{2\gamma+1}}), \gamma = k + a$$

Although at first glance the theorem might seem to make some strong assumptions, notice that it also holds when f is not differentiable, in the special case where $k = 0$. If we also set $a = 1$ we can see that it holds for functions who are simply Lipschitz continuous. Thus, it does not require differentiability of functions $\Delta P(\cdot), \Delta C(\cdot)$. In that case, the theorem provides a $O((N^{-1} \log N)^{\frac{1}{3}})$ convergence rate for the estimated set.

Thus far, we have shown a way to both infer the exact rationalizable set and a good estimation of it using sampling. In practice, we are interested in extracting a single point from that set. A reasonable choice is to select the point which corresponds to the smallest ϵ, i.e. the valuation under which the learner would have the smallest regret. Below are the results from [NST15] who used that approach to infer bidders' valuations using datasets from Bing and found out that advertisers bid approximately 60% less than their true value.

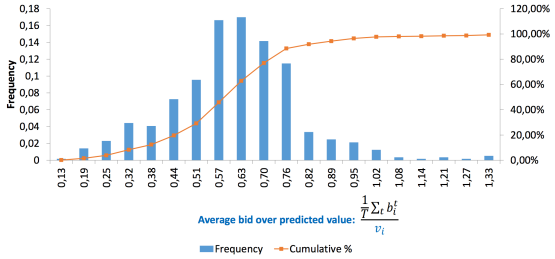


Figure 5.1: Bid Shading

Nisan and Noti [NN17b] used data from an experiment that they conducted, in which actual human beings participated in simulated auctions that were similar to those that are run in practice. Each bidder i was randomly assigned a private value v_i, which in the GV setting was known to him whereas in the DV was unknownm and the valuations of his opponents were not revealed. They found out that as time went by, the average regret of each bidder was decreasing, but the reget of players who had low valuation was much higher than that of players who had high valuation. This is evidence of irrational behavior by low value players. A possible psychological explanation for the previous fact is that when these players behave rationally, they win the lowest ad slots and although their utility is high, they have impulse to rank higher than their opponents, even if that leads to lower utility. In such settings, the point estimation method which we described before is not very effective. The results for the various valuations and auction formats are presented in the following figure.

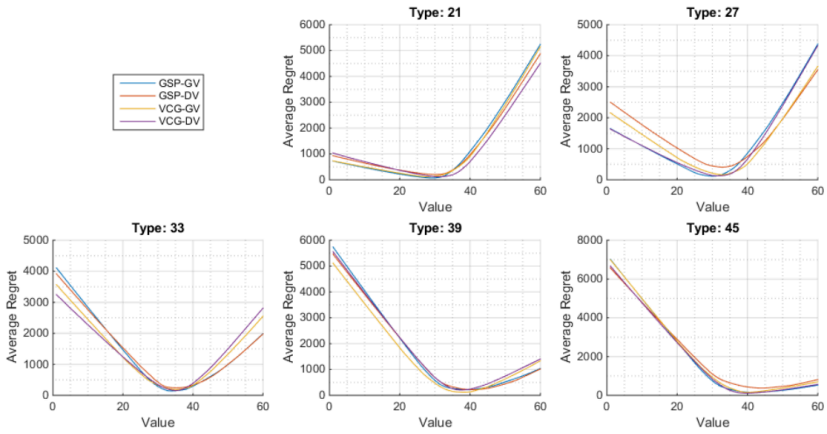


Figure 5.2: [NST15] method evaluation

We can clearly see that the auction format does not have a significant effect on the outcome of the prediction, but the type (value) does. When players have low valuations, the estimation method overshots the actual value, whereas when the type is higher there is an undershooting.

Based on their previous findings, they proposed a different method to predict the private valutation [NN17a]. The drawback of [NST15] projection method is that they use only the tip of the rationalizable set, so it should be modified a little bit in order to use encapsulate more information that is provided from the rationalizable set. Furthermore, learning agents are not able to optimize perfectly their behavior, but they are more likely to act in a way that has low regret. Thus, instead of using only the one with the minimum regret, one natural

extension is to assign weights to each point and output the weighted average. Since we want to assign more weight to valuations that have low regret, we can use a rule similar to MWU algorithm and assign weights which are exponentially decreasing in the regret of that value. Formally, we have that

$$\hat{v} = Z^{-1} \sum_{v} v \cdot e^{-\lambda \sum_{i=1}^{T} regret_i(v)}$$

where Z is a normalization constant and λ is a tuning parameter. Notice that this is actually a generalization of the previous approach, as when $\lambda \to \infty$ the prediction approaches that of [NST15]. This is called the *quantal regret* method. Below are presented the results in the VCG auction of the comparision between the quantal regret method (QR), the min-regret method ([NST15], MR) and a classical approaches (EQ) that assumed equilibrium. The metric that it is used is the root mean square error RMSE $= \sqrt{\frac{1}{|S|} \sum_{\hat{v} \in S} |\hat{v} - v|^2}$, where S is the set of all players that participate in the game.

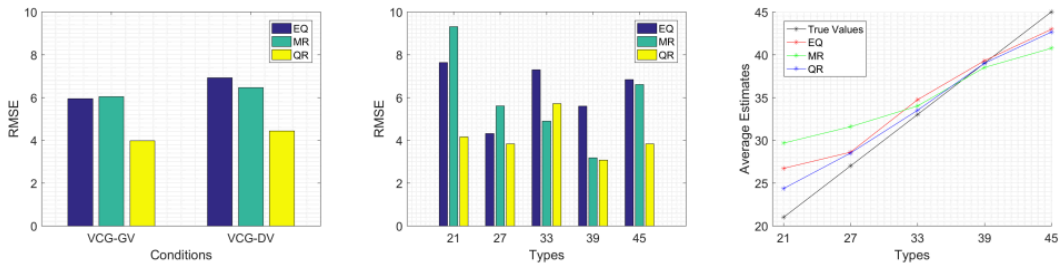


Figure 5.3: Estimation methods comparision under the VCG mechanism

We can see that the QR method outperforms every other. Similar results hold when the GSP auction format is used, which are presented below.

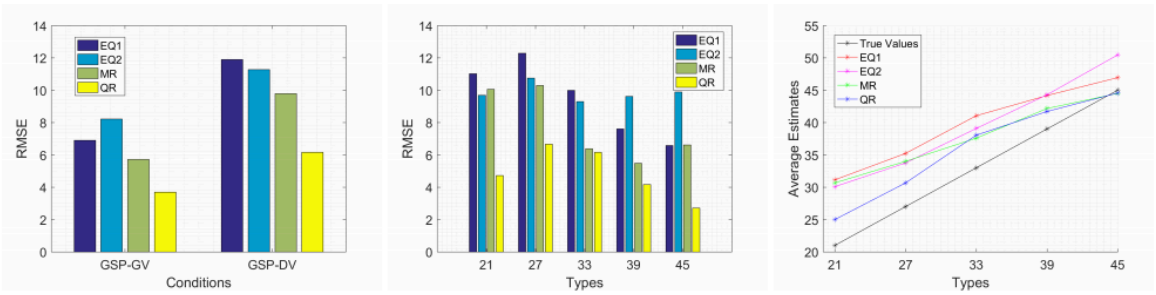


Figure 5.4: Estimation methods comparision under the GSP mechanism

5.2 Efficiency Guarantees

An interesting line of work in the analysis of strategic interactions of learning agents has to do with the efficiency guarantees of the outcome of the game, i.e. how "good" is the result that is achieved for all the players. The most common metric for that quantity is the welfare, which is simply the sum of utilities of all participants. Learning algorithms converge to coarse

correlated and correlated equilibria (we defined these notions of steady states in a previous chapter), therefore it is natural to analyze the welfare in these equilibria. Syrgkanis et al. [SALS15] proved that in a wide class of learning algorithms that have a recency bias the welfare converges to the optimal at the rate $O(1/T)$, meaning that the (expected) welfare of these algorithms is at least $(\lambda/(1+\mu))\text{OPT} - O(1/T)$, where λ, μ are parameters of the smoothness condition of the game, which were introduced by Roughgarden [Rou09]. Furthermore, for these algorithms each player's average (expected) regret converges to zero at the rate $O(T^{-\frac{3}{4}})$. Subesequently, Foster et al. [FLL+16] investigated algorithms that achieve low approximate regret, meaning that the cumulative cost of the learner who uses them multiplicatively approximates the cost of the best action they could have chosen in hindsight. These algorithms form a broader class than those which have a recency bias, this property seems to be ubiquitous among learning algorithms. They improved the previous results in the following ways; the convergence rate was improved by a factor of n (the players in the game), the learning agents required less feedback from the environment in order to achieve the learning task and the convergence occured with high probability (the previous work explored the expected outcome).

There is another interesting work by Lykoyris et al. [LST16], which investigates the efficiency of the outcomes of games with dynamic population, i.e. games in which the players that participate are not fixed over time. In their model, at each time step every player is replaced with an arbitrary new player with probability p (independently), or equivalently each player changes his valuation with probability p. In that setting, the socially optimal solution varies over time and, from a learner's perspective, the *static* regret is too weak of a benchmark. Thus they assume that players use algorithms that achieve no adaptive regret, which is a natural assumption in such a dynamic setting. The intuition behind their results is that in many games that satisfy a "smoothness" property, even when the population is changing, there is an approximate solution that remains relatively stable over time (it is not very sensitive to population variations) and that solution can serve as a benchmark for the sequence of outcomes that take place in the actual game. In many cases, like matching markets, greedy algorithms give stable approximate solutions. If the outcomes are close to that benchmark, then they are also close to the optimum.

Before we state the results we have to provide some definitions.

Definition 5.2.1. A randomized sequence of solutions $\boldsymbol{x}^{1:T} = \{\boldsymbol{x}^1, ..., \boldsymbol{x}^T\}$ and types $\boldsymbol{v}^{1:T} = \{\boldsymbol{v}^1, ..., \boldsymbol{v}^T\}$ is k-stable if the average expected number of changes in each individual player's solution or type is at most k, i.e. if $k_i(v_i^{1:T}, x_i^{1:T})$ is the number of times that $x_i^t \neq x_i^{t+1}$ or $v_i^t \neq v_i^{t+1}$, then

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[k_i(v_i^{1:T}, x_i^{1:T})] \leq k$$

Definition 5.2.2. A cost minimization game G is (λ, μ)-smooth with respect to a solution $\boldsymbol{x}$, if for some $\lambda > 0$ and $\mu < 1$, for any type profile $\boldsymbol{v}$, for each player i there is a strategy $s_i^* \in S_i$ depending on his type v_i and his part of the solution x_i such that for any strategy profile s

$$\sum_i c_i(s_i^*(v_i, x_i), s_{-i}; v_i) \leq C(\boldsymbol{x}; \boldsymbol{v}) + \mu C(s; \boldsymbol{v})$$

We are now ready to present the main theorem regarding cost minimization games.

***Theorem* 5.2.1.** *Consider a repeated cost game with dynamic population $\Gamma = (G, T, p)$ such that the stage game G is solution based (λ, μ) smooth and costs are bounded in $[0, 1]$.*

Suppose that $\boldsymbol{v}^{1:T}, \boldsymbol{x}^{1:T}$ are k-stable sequences and that $\boldsymbol{x}^t$ is a feasible and a-approximately (in expectation) optimal solution for each t, i.e. $\mathbb{E}[C(s^t; \boldsymbol{v}^t)] \leq a\mathbb{E}[OPT(\boldsymbol{v}^t)]$. If players are using an adaptive no regret algorithm with constant C_R then

$$\sum_t \mathbb{E}[C(s^t; \boldsymbol{v}^t)] \leq \frac{\lambda a}{1-\mu} \sum_t \mathbb{E}[OPT(\boldsymbol{v}^t)] + \frac{n}{1-\mu} C_R \sqrt{T(k+1)\ln(NT)}$$

Notice that the first term is reminiscent of the static price of anarchy bounds, multiplied with the approximation ratio of the benchmark that we are competing against, while the second term encodes the learning behavior of the participants. They also proved similar results in other settings, such as matchings markets and badnwidth allocation games.

5.3 Revenue Guarantees

A recent work by Braverman et al. [BMSW18] analyzed single item-single buyer auctions from a revenue point of view. More concretely, they considered a series of single item auctions which have only a single participant. They assume that the bidder is a no-regret learning agent and try to find out how should the seller design the mechanism in order to maximize his revenue, if he knows that bidding behavior from the seller. In their model, at each round t the bidder chooses his valuation v_t randomly (and independently from round to round) from a fixed distribution D, which is known to the seller, submits a bid b_t to the seller, and the seller picks an allocation and a payment function $x_t(\cdot), p_t(\cdot)$, such that $p_t(b) \leq b \cdot x_t(b), \forall t, b$. A simple strategy for the seller, since the valuation distribution is known to him, is to simply use the Myerson's revenue-optimal reserve price and if the bid is higher than that allocate the item and charge the buyer the reserve price, whereas whenever it is lower do nothing. The buyer will learn to bid higher than the reserve price whenever his valuation is higher, and lower whenever it is lower. Thus, the revenue that is generated from that strategy is $T \cdot Rev(D) - o(T)$. The question that arises is whether the seller can do any better than that.

When the bidder uses an algorithm that works like MWU the answer to the previous question is that the seller can do much better than that, in fact he can collect revenue which is arbitrarily close to the welfare. The technique to achieve might seem a little odd; the seller gives away the item for free in the first rounds and then, with a careful transition, charges very high price. Although it might seem unnatural at first glance, their is a clear intuition behind this approach. Recall the way that MWU works, the probability mass gets concentrated very quickly in actions that performed better in the past, thus by giving the item away for free (for carefully chosen high bids) the seller can trick the learning algorithm into putting a lot of mass in those actions. Therefore, when the prices start getting higher, the probability that those high bids will be selected, will remain relatively high for a fairly long amount of time. As it turns out, MWU is not the only algorithm that has this property. We provide the formal definition below.

Definition 5.3.1. Let $\sigma_{it} = \sum_{s=1}^{t} r_{is}$ be the sum of rewards of action i until round t. An algorithm for the experts problem is γ-mean-based if whenever $\sigma it < \sigma_{jt} - \gamma T$ then the probability that the algorithm pulls arm i on round t is at most γ. We say that an algorithm is mean-based, if it is γ-mean-based for some $\gamma = o(1)$.

Intuitively, mean-based algorithms pick actions who have significant less reward so far, only with a small probability. We provide the formal theorem that holds in that case below.

***Theorem* 5.3.1.** *Let $Val(D) = \mathbb{E}_{v\sim D}[v]$. Then, if the buyer is running a mean-based algorithm, for any constant ϵ, there exists a strategy for the seller which obtains revenue a least $(1-\epsilon)Val(D)T - o(T)$.*

In the above theorem, we have a $(1-\epsilon)$ approximation of the welfare and not the exact welfare, since we have to use prices that motivate the bidder to buy the item, so he has to have even a negligible utility. The $o(T)$ term is the price we pay for tricking the learning algorithm and giving away the item for free at the beginning.

We have so far seen how the seller can leverage the learning behavior of the buyer in order to outperform Myerson's revenue. Thus, it is natural to wonder whether the buyer can used a learning algorithm which cannot be fooled in that way. The answer is again positive, but we have to require a stronger notion of no-regret (which is still achievable by starting from an arbitrary no-regret algorithm and using a black-box construction). The learning algorithm needs to have to regret not only against every fixed bid, but also with respect to the "policy of play" as if the learner had any lower value v' than his current. This reduction looks like the one used to achieve no-swap-regret, we take several copies of the learning algorithms who ran for different valuations v' and we carefully tune them to achieve the learning task. Under this learning behavior, the following theorem holds for the seller.

***Theorem* 5.3.2.** *There exists a no-regret learning algorithm for the buyer against which every selling strategy extracts no more than $Mye(D)T + O(m\sqrt{\delta T})$ revenue, where $Mye(D)$ is the revenue generated from using Myerson's reserve price.*

The intuition behind the previous theorem is the following; the key property of the learning behavior which is "not regretting playing as if my value were v'" looks like "not preferring to report v' instead of v", which suggests that the average allocation probabilities and prices paid by the buyer using that algorithm should look like the ones in a truthful auction. Thus, the average revenue cannot exceed that of Myerson's pricing strategy.

It is worth investigating auctions in which overbidding is a dominated strategy, which arise many times. We say that an auction is *critical* whenever the previous property holds, and the buyer is *clever* if he never plays a dominated bid. Suppose that the buyer is clever and he simply uses a mean-based algorithm (not something better like we discussed before). Then, the seller can extract revenue $MBRev(D) \cdot T$, which is tight, where $MBRev(D)$ is the solution of the following linear program.

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} q_i(v_i x_i - u_i) \\ \text{subject to} \quad & u_i \geq (v_i - v_j)x_j \quad , \forall i, j \in [m], i > j \\ & u_i \geq 0, 1 \geq x_i \geq 0, \forall i \in [m] \end{aligned}$$

We will now explain the LP that LP. For the function that we want to maximize, q_i is the probability that the buyer has valuation v_i, x_i represents the average probability that the learner gets the item when his has value v_i and u_i is his average utility under that valuation. Thus, the bidder's average value is $v_i x_i$ so the price that they pay is $v_i x_i - u_i$, therefore we can see that the objective we are maximizing is the revenue. The second line of constaints ensure simply a normalization of the values. The first line of constraints look like those in a truthful auction; the LHS is the utility of the buyer with value v_i for telling the truth, but his utility for reporting v_j is $(v_i - v_j)x_j + u_j$, so the term u_j is missing. The outline of the proof is the following.

- The buyer has no regret when he has value v_i, so his utility must be at least as high as playing arm j every time.
- The auction never charges arm j more than v_j (if it wins the item), the buyer's utility for playinr arm j every round is at least $y_j(v_i - v_j)$, where y_j is the average probability that arm j wins.
- The auction is monotone and the buyer never considers overbidding, thus if x_j is the probability that he gets the item when he has value v_j it holds that $y_j \geq x_j$.

The following two theorems hold, regarding $MBRev(D)$.

***Theorem* 5.3.3.** *Any strategy for the seller achieves revenue at most $MBRev(D)T + o(T)$ against a buyer who runs a no-regret learning algorithm and never overbids.*

***Theorem* 5.3.4.** *For any $\epsilon > 0$ there exists a strategy for the seller that gets revenue at least $(MBRev(D) - \epsilon)T - o(T)$ against a buyer who runs a mean-based algorithm who overbids with probability 0. The strategy sets a decreasing cutoff r_t and for all rounds t awards the item only whenever $b_t \geq r_t$.*

Thus, we see that the revenue the seller collects in that case is tight.

Chapter 6

Results

In this chapter we will present an extension of the [NST15] inference method that we discussed previously, in an environment where the valuations of the bidders vary over time. Valuations represent the average income that will be generated for the advertiser once a user clicks on his ad. Therefore it is reasonable to expect that these valuations will not remain constant, but they will fluctuate slowly, since they are affected by many factors, including seasonality trends. For instance, on Valentine's Day flower shops' owners will value their ads higher than the day before, as users are much more likely to convert these clicks to actual purchases.

Moreover, we will demonstrate a way to use the inference method suggested by [NST15] in the single buyer-single item setting, when the valuation of the bidder remains constant, in order to maximize the seller's revenue. We do not make any assumptions about the quality of the prediction and we bound the total revenue that is lost by a function of the difference of the initial prediction and the actual valuation.

6.1 Inferring Time-Varying Valuations

6.1.1 Model

We consider the problem of inferring time varying valuations of bidders who participate in online keyword auctions, where at each time step they submit a single dimensional bid. We assume that bidders have quasilinear utilities that vary over time, namely $u_t(b) = v_t \cdot P_t(b) - C_t(b)$, where $v_t, P_t(\cdot), C_t(\cdot)$, are the bidder's valuation, click probability function and cost per click function respectively, at time t. At each step we observe the bid b_t as well as the aforementioned functions. We assume that $P_t(\cdot)$ is concave and $C_t(\cdot)$ is convex, which is true in several contexts of interest, such as the GSP mechanism. Moreover, we assume that bidders are learning agents who achieve *no dynamic regret*, meaning that as time goes by their average utility per time step is as good the best sequence of bids. More concretely, they achieve the following objective:

$$\frac{1}{T}(\sum_{t=1}^{T} v_t P_t(b_t) - C_t(b_t)) \geq \frac{1}{T}(\sum_{t=1}^{T} v_t P_t(b'_t) - C_t(b'_t)) - \epsilon, \forall \boldsymbol{b'} \in B^T \tag{6.1}$$

where B denotes the space of all possible bids, $\boldsymbol{b}$ is the learner's bid sequence and ϵ denotes the regret, which, as $T \to \infty$, goes to 0 (in the whole text bold letters denote vectors). Although dynamic regret might seem too strong of a benchmark, there are many algorithms that achieve it, for slowly changing utility functions. Intuitively, the reason why we need the learner to achieve that instead of the weaker *no static regret* benchmark which compares the performance of the learner to that of the best *fixed* bid in hindsight, is that we want the learning process to be biased towards more recent valuations and click probability, cost

functions, so that the bids correspond better to the current valuation and functions, instead of converging to some kind of "average".

6.1.2 Rationalizable Set

Let $\boldsymbol{v} = (v_1, ..., v_T)$ be a valuation sequence and $\boldsymbol{b} = (b_1, .., b_T)$ be the bid sequence that the learner actually played. We define the rationalizable set NR to be the set of all $(\boldsymbol{v}, \epsilon)$ tuples that satisfy inequality (1). Let $\Delta P_t(b'_t) = \frac{1}{T}(P_t(b'_t) - P_t(b_t)), \Delta C_t(b'_t) = \frac{1}{T}(C_t(b'_t) - C_t(b_t))$, which measure the (normalized) change in click probability and cost when switching from the observed bid b_t to b'_t at time t, and $\boldsymbol{\Delta P(b')} = (\Delta P_1(b'_1), ..., \Delta P_T(b'_T)), \boldsymbol{\Delta C(b')} = (\Delta C_1(b'_1), ..., \Delta C_T(b'_T))$. By simply rearranging terms, inequality (1) becomes:

$$\sum_{t=1}^{T} v_t \cdot \frac{1}{T}(P_t(b'_t) - P_t(b_t)) - \sum_{t=1}^{T} \frac{1}{T}(C_t(b'_t) - C_t(b_t)) \leq \epsilon, \forall \boldsymbol{b'} \in B^T$$

$$\boldsymbol{v \cdot \Delta P(b') - 1 \cdot \Delta C(b') \leq \epsilon, \forall b' \in B^T} \tag{6.2}$$

***Lemma* 6.1.1.** *The rationalizable set NR is convex.*

Proof. Let $(\boldsymbol{v}, \epsilon) \in NR$ and $(\boldsymbol{v'}, \epsilon') \in NR$. Then, by inequality (2), for an arbitrary $\boldsymbol{b'} \in B^T$ we get that $\boldsymbol{v \cdot \Delta P(b') - 1 \cdot \Delta C(b') \leq \epsilon}$, $\boldsymbol{v' \cdot \Delta P(b') - 1 \cdot \Delta C(b') \leq \epsilon'}$. Multiplying these by $\lambda, 1-\lambda$ respectively to take the convex combination of the two points, we get that $\lambda \boldsymbol{v \cdot \Delta P(b') - \lambda 1 \cdot \Delta C(b') \leq \lambda \epsilon}$, $(1-\lambda)\boldsymbol{v' \cdot \Delta P(b') - (1-\lambda) 1 \cdot \Delta C(b') \leq (1-\lambda)\epsilon'}$. Finally, by adding them together we get that $(\lambda \boldsymbol{v} + (1-\lambda)\boldsymbol{v'}) \cdot \boldsymbol{\Delta P(b') - 1 \cdot \Delta C(b') \leq \lambda \epsilon + (1-\lambda)\epsilon'}$. Since $\boldsymbol{b'}$ was arbitrary this holds $\forall \boldsymbol{b'} \in B^T$, so $(\lambda \boldsymbol{v} + (1-\lambda)\boldsymbol{v'}, \lambda \epsilon + (1-\lambda)\epsilon')$, which is the convex combination of the two starting points, is indeed in NR. □

6.1.3 Point Prediction

We are interested in predicting a meaningful point from that set and in order to do so we have constructed a Linear Program. Let's start by presenting the intuition behind it. Firstly, as time goes by, the learner's average dynamic regret goes to 0. Furthermore, no matter which valuation sequence we answer, the average dynamic regret of that sequence will always be lower bounded by 0, since the benchmark is the optimal bid sequence. So it is reasonable to answer the point with the minimum dynamic regret, since the learner's actual regret converges to the regret of that point. If we view answering the right $\boldsymbol{v}$ as a data fitting problem, where we need to find a valuation sequence that best explains the learner's decisions, and by explains we mean minimizes their dynamic regret for the bid sequence we have observed, we have the following tradeoff: if we allow the sequence to be very "expressive", and by that we mean vary very much between consecutive time steps, it will probably "overfit", meaning that it will try to justify bid changes which might be caused from other factors, such as changes in $P_t(\cdot)$ or $C_t(\cdot)$, as changes in valuation in order to reduce the regret that the learner would presumably have if their actual valuation sequence was the one we answer. In order to avoid that, we restrict the expressibility of our class of valuation sequences by imposing a Lipschitz alike condition $|v_t - v_{t-1}| \leq k, t \in \{2, .., T\}$, where k is a constant upper bound. Thus, we

have the following LP:

$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{T} \epsilon_t \\ \text{subject to} \quad & \tfrac{1}{T}(v_t \Delta P(b') - \Delta C(b')) \leq \epsilon_t \ , t = 1, ..., T, b' = b_1, ..., b_{|B|} \\ & v_t - v_{t-1} \leq k, v_{t-1} - v_t \leq k, t = 2, ..., T \end{aligned}$$

Each of the $|B|$ consecutive constraints from the first $|B| \cdot T$ measure the regret on that round and in order to minimize the total regret we simply have to minimize the regret on each round. The last $2T - 2$ constraints control the variance of the answer. So, in total, the $2T$ decision variables are $v_1, ..., v_T, \epsilon_1, ..., \epsilon_T$ and we have $|B| \cdot T + 2T - 2$ constraints. If the upper bound on the variance isn't known a priori, we can minimize the function $\lambda \sum_{t=1}^{T} \epsilon_t + (1-\lambda)k$, where λ is a free variable.

6.1.4 Magnitude of changes allowed

The bidder's utility depends on their valuation and on the click probability and cost per click functions, which in turn depend on the mechanism and the behaviour of the other players. We present some bounds on the allowed variation of the bidder's valuations, as well as the aforementioned functions, in order to achieve vanishing dynamic regret. Let $V_T = \sum_{t=2}^{T} \sup_{x \in X} |f_t(x) - f_{t-1}(x)|$, Besbes et al. [BGZ15] established regret bounds of the form $\sum_{t=1}^{T} \mathbb{E}[f_t(x_t)] - f_t(x_t^*) = O(T^{2/3}(1 + V_T)^{1/3})$. This means that the average regret of a learner who uses that algorithm is $O(T^{-1/3}(1 + V_T)^{1/3}) = O((1/T + V_T/T)^{\frac{1}{3}})$, so in order to have vanishing regret it should hold that $V_T = o(T)$. We have the following lemma.

***Lemma* 6.1.2.** *Let $\epsilon_{P_t(x)} = P_t(x) - P_{t-1}(x), \epsilon_{C_t(x)} = C_t(x) - C_{t-1}(x)$. If $v_t \leq V, \forall t$, in order for the learner to have vanishing dynamic regret the following conditions should hold:*
- $\sum_{t=2}^{T} |v_t - v_{t-1}| = o(T)$
- $\sum_{t=2}^{T} \sup_{x \in X} |\epsilon_{P_t(x)}| = o(T)$
- $\sum_{t=2}^{T} \sup_{x \in X} |\epsilon_{C_t(x)}| = o(T)$

Proof. The proof of the above lemma follows by simply noticing that we can relate the deviation of the learner's utility to the variation of his valuation and the functions that he is provided by the mechanism.

$|f_t(x) - f_{t-1}(x) = v_t P_t(x) - C_t(x) - v_{t-1} P_{t-1}(x) + C_{t-1}(x)| \leq$

$|v_t(P_{t-1}(x) + P_t(x) - P_{t-1}(x)) - v_{t-1} P_{t-1}(x)| + |C_t(x) - C_{t-1}(x)| =$

$|v_t(P_{t-1}(x) + \epsilon_{P_t(x)}) - v_{t-1} P_{t-1}(x)| + |C_t(x) - C_{t-1}(x)| =$

$|P_{t-1}(x)(v_t - v_{t-1}) + v_t \epsilon_{P_t(x)}| + |C_t(x) - C_{t-1}(x)| \leq$

$|P_{t-1}(x)(v_t - v_{t-1})| + |v_t \epsilon_{P_t(x)}| + |C_t(x) - C_{t-1}(x)| \leq$

$|v_t - v_{t-1}| + V|\epsilon_{P_t(x)}| + |\epsilon_{C_t(x)}|$

So, $V_T \leq \sum_{t=2}^{T} |v_t - v_{t-1}| + V \sum_{t=2}^{T} \sup_{x \in X} |\epsilon_{P_t(x)}| + \sum_{t=2}^{T} \sup_{x \in X} |\epsilon_{C_t(x)}|$, which implies that if $\sum_{t=2}^{T} |v_t - v_{t-1}| = o(T), \sum_{t=2}^{T} \sup_{x \in X} |\epsilon_{P_t(x)}| = o(T), \sum_{t=2}^{T} \sup_{x \in X} |\epsilon_{C_t(x)}| = o(T)$ then $V_T \leq o(T)$.

□

6.1.5 Simple Simulations

We present the results of some simple simulations in which a learner uses a variation of gradient descent and at each round is given some slowly changing click probability and cost functions, which are concave and convex respectively. The click probability function is $P(b) = \sqrt{b} + \epsilon b$, where ϵ is some small random noise and the cost function is $C(b) = \frac{1}{2}b^2 + \epsilon' b$, where ϵ' is again some small random noise. The green curve represents the learner's bids, the orange represents the actual valuations and the blue the inferred valuations. The valuations are in the interval $[0, 1]$ and we have run the simulation for 300 iterations.

We first present the results of the LP in which a constant upper bound on the variance of the valuations is known a priori.

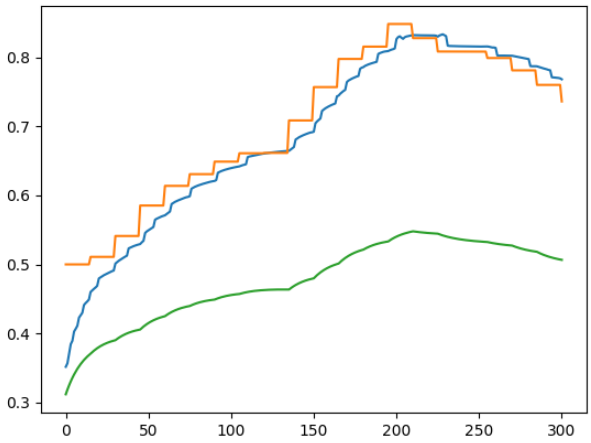


Figure 6.1: Inference with known bound on the variance

Below we present the results of the LP in which the upper bound is not known (the second version which we presented above).

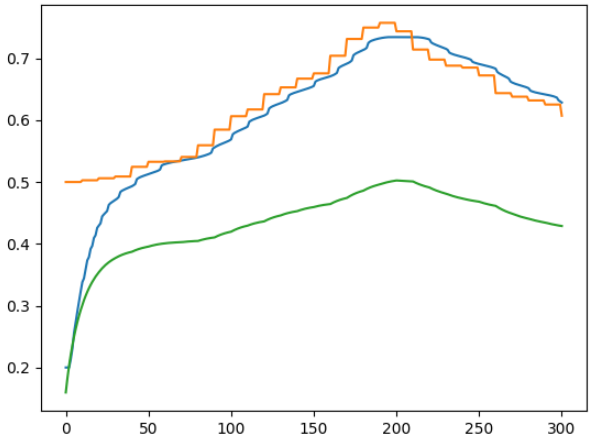


Figure 6.2: Inferenece with unknown upper bound on the variance

6.2 Maximizing Revenue in Single Buyer-Single Item Auctions

We are interested in using the valuation inference method in a setting where we have a single bidder bidding for a single item over a period of T iterations, and whose valuation v for that item remains constant over time in order to maximize the seller's revenue. In that setting, the only way to affect the auctioneer's revenue, is through using carefully chosen reserve prices. The rules of the game are simple; at each time step we set a reserve price p_t and if $b_t \geq p_t$ then the bidder gets the item and pays p_t, whereas if $b_t < p_t$ the bidder does not get the item and of course he does not pay anything. Since we do not know anything about the quality of the prediction beforehand, we cannot simply set the price close to that prediction. Instead, we will do a binary search using that prediction as a starting point. We assume that the buyer is a learning agent who has $O(\sqrt{T})$ cumulative regret, namely

$$\sum_{t=1}^{T} u_t(b^*) - \sum_{t=1}^{T} u_t(b_t) \leq a\sqrt{T}, \forall b^* \in B \tag{6.3}$$

where $u_t(b_t) = (v - p_t)\mathbb{1}_{\{b_t > p_t\}}$ and a is some known constant. Note that even if we set a selling price lower than his actual valuation, we do not know for sure that the player will buy the item, we have to let some time pass in order to allow him to "learn" how to play under this reserve price.

Let v^* be the actual valuation of the learner. Then $\sum_{t=1}^{T} u_t(b_t) \leq v^*T, \forall (b_1, ..., b_T) \in B^T$. Since this auction is truthful we know that $u_t(v^*) \geq u_t(b), \forall b \in B, \forall t \in [T]$.

***Lemma* 6.2.1.** *If we set a price p for $c\sqrt{T}$, with $c \geq a$, iterations and the player does not buy the item then we have that $v^* \leq p + \frac{a}{c}$. If he buys it, then $v^* \geq p$ since we assume that he never bids higher than his valuation.*

Proof. For contradiction, assume that although the learner does not buy this item in the $c\sqrt{T}$ iteration under price p, it holds that $v^* > p + \frac{a}{c}$. Then $\text{regret}_T = \sum_{t=1}^{T} u_t(v^*) - u_t(b_t) \geq c\sqrt{T}(v^* - p) > c\sqrt{T}(p + \frac{a}{c} - p) = aT$, which is a contradiction since we know that $\text{regret}_T \leq aT$. □

The above lemma gives us a simple selling strategy. We assume that the bidding space is discrete and that the difference between consecutive bids is b_0. Let $\eta = \frac{v_0 - v^*}{b_0}$, where v_0 is the point that the estimation method predicts, and for simplicity of the exposition let $\frac{a}{c} = \epsilon$. The intuition is that instead of searching at the whole valuation space V, we can construct an interval of length $\Theta(\eta)$ that is guaranteed to contain the actual valuation and do a binary search on that space, so that whenever our prediction is good the search will be faster. Our initial selling price is $p_0 = v_0$ which we keep for $c\sqrt{T}$ iterations and depending on whether the player buys the item or not we set the price to be $v_0 + 2b_0, v_0 + 4b_0, ...$ or $v_0 - 2b_0, v_0 - 4b_0, ...$ respectively. Every time we change the price, we keep it for $c\sqrt{T}$ iterations. We stop this procedure when the outcome of the auction differs for the first time from the initial outcome, i.e. if the player did not buy at the beginning we stop whenever he buys for the first time and vice versa. Thus, after $\Theta(\log \eta)$ iterations, we have an interval of length $\Theta(\eta)$, which we know for sure that contains v^*. We then continue by doing a binary search on that interval until it has size ϵ (we still insist on every price for $c\sqrt{T}$ iterations). There is a small asymmetry in the search since whenever the player buys the item on price p we know that $v^* \geq p$, whereas if he does not buy it we have that $v \leq p + \epsilon$, but it does not make a big difference. In the worst case, the search interval has lengths $\frac{\Theta(\eta)+\epsilon}{2}, \frac{\Theta(\eta)+3\epsilon}{4}, \frac{\Theta(\eta)+7\epsilon}{8}, ..., \frac{\Theta(\eta)+(2^{i-1}+1)\epsilon}{2^i}$. We want after i rounds the interval to have length at most ϵ, thus the number of rounds i that are

needed are $\frac{\Theta(\eta)+(2^{i-1}+1)\epsilon}{2^i} \leq \epsilon \implies i \geq \log \frac{\Theta(\eta)}{\epsilon} + 1$. The revenue that is generated from the procedure described before is at least $(v^* - \epsilon)T - \Theta((\log \eta + \log \frac{\eta}{\epsilon})\sqrt{T})$.

6.3 Future Work

In this thesis, we have made the first step towards inferring time-varying private valuations in single-parameter environments. Although our results seem promising there is still work to be done in that direction. The main future directions, in our opinion, are the following.

- Since we have to answer a sequence of valuations, the rationalizable set seems to expand very quickly and it consists of valuation sequences that do not seem to be rational. We have imposed a Lipschitz-alike condition in order to answer meaningful sequences, but we believe that there are more constraints that can be taken into account so that the resulting projection of the set is a better estimation of the actual one.
- A very interesting direction, which we believe that is tightly connected to the one that we mentioned above, is to find a way to use the valuation sequence that our estimation method predicts in order to set the right reserve prices and maximize the revenue in a dynamic environment. There is a key challenge that we need to tackle in order to solve that problem; if we insist on a reserve price for a long period of time the valuation of the player might change rapidly whereas if we set it only for a few iterations it might be the case that the bidder will not manage to learn how to play correctly. By setting the reserve prices correctly we can reduce the size of the rationalizable set because many potential valuations can be ruled out, as they lead to a very high regret.

Bibliography

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[AN10] Susan Athey and Denis Nekipelov. A structural model of sponsored search advertising auctions. In *Sixth ad auctions workshop*, volume 15, 2010.

[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.

[BGZ15] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

[BHN13] Patrick Bajari, Han Hong, and Denis Nekipelov. Game theory and econometrics: A survey of some recent research. In *Advances in economics and econometrics, 10th world congress*, volume 3, pages 3–52, 2013.

[BMSW18] Mark Braverman, Jieming Mao, Jon Schneider, and Matt Weinberg. Selling to a no-regret buyer. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 523–538. ACM, 2018.

[Bro51] George W Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.

[BVN50] George W Brown and John Von Neumann. Solutions of games by differential equations. Technical report, RAND CORP SANTA MONICA CA, 1950.

[CKKK11] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. On the efficiency of equilibria in generalized second price auctions. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 81–90. ACM, 2011.

[CYL+12] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, pages 6–1, 2012.

[DGP09] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[EOS07] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review*, 97(1):242–259, 2007.

[FLL+16] Dylan J Foster, Zhiyuan Li, Thodoris Lykouris, Karthik Sridharan, and Eva Tardos. Learning in games: Robustness of fast convergence. In *Advances in Neural Information Processing Systems*, pages 4734–4742, 2016.

[H+16] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[HS07] Elad Hazan and Comandur Seshadhri. Adaptive algorithms for online decision problems. In *Electronic colloquium on computational complexity (ECCC)*, volume 14, 2007.

[HTW18] Darrell Hoy, Sam Taggart, and Zihe Wang. An improved welfare guarantee for first price auctions. *arXiv preprint arXiv:1803.06707*, 2018.

[HW15] Eric C Hall and Rebecca M Willett. Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):647–662, 2015.

[JLB07] Albert Xin Jiang and Kevin Leyton-Brown. Bidding agents for online auctions with hidden bids. *Machine Learning*, 67(1-2):117–143, 2007.

[JNT17] Pooya Jalaly, Denis Nekipelov, and Eva Tardos. Learning and trust in auction markets. *arXiv preprint arXiv:1703.10672*, 2017.

[JRSS15] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Artificial Intelligence and Statistics*, pages 398–406, 2015.

[KM08] David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. In *International Workshop on Internet and Network Economics*, pages 585–596. Springer, 2008.

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[LST16] Thodoris Lykouris, Vasilis Syrgkanis, and Éva Tardos. Learning and efficiency in games with dynamic population. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 120–129. Society for Industrial and Applied Mathematics, 2016.

[LW94] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[MSJR16] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. *arXiv preprint arXiv:1603.04954*, 2016.

[Mye81] Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

[Nas51] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[NN17a] Noam Nisan and Gali Noti. A" quantal regret" method for structural econometrics in repeated games. *arXiv preprint arXiv:1702.04254*, 2017.

[NN17b] Noam Nisan and Gali Noti. An experimental evaluation of regret-based econometrics. In *Proceedings of the 26th International Conference on World Wide Web*, pages 73–81. International World Wide Web Conferences Steering Committee, 2017.

[NRTV07] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.

[NST15] Denis Nekipelov, Vasilis Syrgkanis, and Eva Tardos. Econometrics for learning agents. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 1–18. ACM, 2015.

[Rob51] Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

[Rou09] Tim Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 513–522. ACM, 2009.

[Rou12] Tim Roughgarden. The price of anarchy in games of incomplete information. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 862–879. ACM, 2012.

[Rou16] Tim Roughgarden. *Twenty lectures on algorithmic game theory*. Cambridge University Press, 2016.

[RS13a] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. 2013.

[RS13b] Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

[SALS15] Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems*, pages 2989–2997, 2015.

[SS+12] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[ST13] Vasilis Syrgkanis and Eva Tardos. Composable and efficient mechanisms. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 211–220. ACM, 2013.

[Var07] Hal R Varian. Position auctions. *international Journal of industrial Organization*, 25(6):1163–1178, 2007.

[Vic61] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[Zin03] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.