Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Εργαστηριο Συστηματων Τεχνητης Νοημοσυνης και Μαθησης

Modeling and Analyzing Musical Structure through Chord Progressions using Statistical and Machine Learning Methods

Diploma Thesis

by

Ioannis Liolitsas

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2025

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Modeling and Analyzing Musical Structure through Chord Progressions using Statistical and Machine Learning Methods

Diploma Thesis

by

Ioannis Liolitsas

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11η Νοεμβρίου, 2025.

.....	.....	.....
Γεώργιος Στάμου	Αθανάσιος Βουλόδημος	Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Ομότιμος Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2025

.....

Ιωαννης Λιολιτσας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η παρούσα διπλωματική εργασία διερευνά τη μοντελοποίηση και ανάλυση της μουσικής δομής μέσω συμβολικών διαδοχών συγχορδιών. Συνδυάζει στατιστικές, λανθάνοντος χώρου και ακολουθιακές προσεγγίσεις μοντελοποίησης, με σκοπό την μελέτη των αρμονικών σχέσεων ανάμεσα σε μουσικά είδη, δεκαετίες και τμήματα τραγουδιών. Χρησιμοποιώντας το μεγάλης κλίμακας σύνολο δεδομένων Chordonomicon, αυτή η εργασία εξετάζει τόσο ερμηνεύσιμες αναπαραστάσεις όσο και αρχιτεκτονικές για πρόβλεψη, δίνοντας έμφαση στην τμηματοποίηση σε επίπεδο τμημάτων και αποσυντεθειμένες αναπαραστάσεις συγχορδιών (ρίζα, ποιότητα + επεκτάσεις, μπάσο) για την αποτύπωση λεπτομερών αρμονικών σχέσεων.

Το μεθοδολογικό πλαίσιο χωρίζεται σε δύο μέρη. Η πρώτη διαδικασία εφαρμόζει στατιστική ανάλυση ακολουθιών n-στοιχείων και μοντέλα λανθάνουσας απόστασης για να ενσωματώσει τις διαδοχές συγχορδιών σε έναν συνεχή χώρο, παρέχοντας μια ερμηνεύσιμη οπτικοποίηση μουσικά σημαντικών μοτίβων για την ταξινόμηση ανά μουσικό είδος και δεκαετία. Για να συμπληρωθεί αυτή η προσέγγιση χρησιμοποιούνται επίσης ακολουθιακά μοντέλα για τη σύγκριση των αποτελεσμάτων. Η δεύτερη διαδικασία υιοθετεί αναδρομικά νευρωνικά δίκτυα, τις παραλλαγές τους με πύλες και μοντέλα κατάστασης-χώρου για την πρόβλεψη της αρμονικής ροής των διαδοχών συγχορδιών. Αυτή η μέθοδος αξιολογεί την επίδραση του μεγέθους του συνόλου εκπαίδευσης, του μήκους τη ακολουθίας που είναι διαθέσιμη πριν από την πρόβλεψη, των αποσυνθέσεων συγχορδιών και των συμφραζομένων σε επίπεδο τμημάτων στην απόδοση των μοντέλων.

Οι στατιστικές (περιγραφικές) αναλύσεις αποκαλύπτουν ισχυρές αρμονικές κανονικότητες, σταθερότητα δομής των τραγουδιών σε όλα τα είδη και τις δεκαετίες, καθώς και συστηματικές παραλλαγές στις αλλαγές συγχορδιών μεταξύ των τμημάτων των τραγουδιών. Για τα προβλήματα ταξινόμησης, τα μοντέλα λανθάνουσας απόστασης παρέχουν ερμηνεύσιμες αναπαραστάσεις που αποτυπώνουν εν μέρει τις αρμονικές και στυλιστικές σχέσεις, ενώ οι ακολουθιακές αρχιτεκτονικές συμπληρώνουν αυτά τα ευρήματα με τη μοντελοποίηση χρονικών εξαρτήσεων. Συνολικά, η διακριτική ικανότητα όλων των μοντέλων παραμένει περιορισμένη, αντανακλώντας τις κοινές αρμονικές αρχές που καθοδηγούν τις στυλιστικές και χρονικές παραλλαγές.

Η εργασία αυτή καταδεικνύει ότι η ενσωμάτωση ερμηνεύσιμων στατιστικών μοντέλων με αποδοτικές ακολουθιακές αρχιτεκτονικές παρέχει σημαντικές πληροφορίες για τη δομή της συμβολικής μουσικής. Τα αποτελέσματα υπογραμμίζουν τη σημασία των συμφραζόμενων σε επίπεδο τμημάτων, της αποσύνθεσης συγχορδιών και της αρμονικής αναπαράστασης για την κατανόηση των οργανωτικών αρχών της τονικής μουσικής. Αυτά τα ευρήματα συμβάλλουν στον τομέα της ανάκτησης μουσικών πληροφοριών και βοηθούν στην κατανόηση των προβλημάτων ταξινόμησης και πρόβλεψης, προσφέροντας μια βάση για μελλοντική έρευνα στην υπολογιστική ανάλυση της μουσικής, τον χαρακτηρισμό του στυλ και την αυτόματη σύνθεση.

Λέξεις κλειδιά — Ανάλυση Συμβολικής Μουσικής, Chordonomicon, Διαδοχές Συγχορδιών, Ανάκτηση Μουσικής Πληροφορίας (ΑΜΠ), Αναδρομικά Νευρωνικά Δίκτυα (ΑΝΔ), Δίκτυα Μακράς και Βραχείας Μνήμης (ΔΜΒΜ), Αναδρομικές Μονάδες με Πύλες (ΑΜΜΠ), Αρμονική Δομή, Ταξινόμηση σε Μουσικά Είδη, Ταξινόμηση σε Μουσικές Δεκαετίες, Πρόβλεψη Συγχορδίας, Τμηματοποίηση ανά Μουσικό Τμήμα, Μοντέλα Κατάστασης-Χώρου, Mamba

Abstract

This thesis investigates the modeling and analysis of musical structure through symbolic chord progressions. It integrates statistical, latent-space, and sequential modeling approaches to study harmonic relationships across genres, decades, and song sections. By utilizing the large-scale Chordonomicon dataset, this work examines both interpretable representations and predictive architectures, with emphasis on part-level segmentation and decomposed chord representations (root, quality+extensions, bass) for capturing fine-grained harmonic relationships.

The methodological framework is split in two parts. The first pipeline applies statistical n-gram analysis and latent distance models to embed chord progressions into a continuous space, providing an interpretable visualization of musically meaningful patterns for genre and decade classification. To complement this approach sequential models are also employed to compare the results. The second pipeline adopts recurrent neural networks, their gated variants, and state-space models to predict the harmonic flow of chord sequences. This method assesses the impact of training set size, the length of the sequence available before prediction, chord decompositions and part-level context on performance.

Descriptive analyses reveal strong harmonic regularities, stability of song structure across genres and decades, and systematic variations in chord transitions across song segments. For classification tasks, latent distance models provide interpretable representations that partially capture harmonic and stylistic relationships, while sequential architectures complement these findings by modeling temporal dependencies. Overall, the discriminative power across all models remains limited, reflecting the shared harmonic principles drive stylistic and temporal variations.

This work demonstrates that integrating interpretable statistical modeling with efficient sequential architectures provides meaningful insights into the structure of symbolic music. The results highlight the significance of part-level context, chord decomposition, and harmonic representation in capturing the organizational principles of tonal music. These findings contribute to the field of music information retrieval and aid in understanding classification and predictive tasks, offering a foundation for future research in computational music analysis, stylistic characterization and automatic composition.

Keywords — Symbolic Music Analysis, Chordonomicon, Chord Progressions, Music Information Retrieval (MIR), Recurrent Neural Networks (RNNs), Long Shot-Term Memory Networks (LSTMs), Gated Recurrent Units (GRUs), Harmonic Structure, Genre Classification, Decade Classification, Chord Prediction, Part-Level Segmentation, State-Space Models, Mamba

Ευχαριστίες

Η παρούσα διπλωματική εργασία δεν θα ήταν δυνατή χωρίς την πνευματική συμβολή πολλών ανθρώπων. Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου, κ Γεώργιο Στάμου, για την ευκαιρία εκπόνησης αυτής της διπλωματικής, για την καθοδήγησή του και για την βοήθεια στην επιλογή ενός θέματος που με ενθουσίασε και με προκάλεσε να δώσω τον καλύτερό μου εαυτό.

Ευχαριστώ επίσης τα μέλη του εργαστηρίου, Βασίλη Λυμπεράτο και Σπύρο Κανταρέλη, για την προθυμία και διαθεσιμότητά τους να λύσουν κάθε απορία μου καθώς και την καθοδήγησή τους που με στήριξε ώστε να αντιμετωπίσω ό,τι προβλήματα προέκυψαν κατά την εκπόνηση της εργασίας, συμβάλοντας καθοριστικά στη διαμόρφωση και βελτίωση του αποτελέσματός της.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου, για την αγάπη, τη στήριξη και την πίστη τους σε εμένα. Η υποστήριξή και οι συμβουλές τους υπήρξαν πηγή δύναμης και έμπνευσης καθ'όλη τη διάρκεια αυτής της ακαδημαϊκής προσπάθειας και χωρίς αυτούς η ολοκλήρωση της δεν θα ήταν δυνατή.

Λιολίτσας Ιωάννης, Νοέμβριος 2025

Contents

Contents 11

List of Figures 14

1 Εκτεταμένη Περίληψη στα Ελληνικά 1
- 1.1 Θεωρητικό Υπόβαθρο . . . 2
  - 1.1.1 Μουσική Θεωρία . . . 2
  - 1.1.2 Ακολουθιακά Μοντέλα . . . 4
  - 1.1.3 Μοντέλα Λανθάνουσας Απόστασης . . . 8
- 1.2 Πρόταση . . . 10
  - 1.2.1 Κίνητρα και ερευνητικοί στόχοι . . . 10
  - 1.2.2 Επισκόπηση του προτεινόμενου πλαισίου . . . 11
- 1.3 Σύνολο Δεδομένων και Ανάλυση . . . 13
  - 1.3.1 Επισκόπηση Συνόλου Δεδομένων . . . 13
  - 1.3.2 Εξαγωγή Χαρακτηριστικών . . . 13
  - 1.3.3 Περιγραφική Ανάλυση . . . 14
  - 1.3.4 Ανάλυση μέσω Μοντέλου Λανθάνουσας Απόστασης . . . 24
- 1.4 Πειράματα . . . 28
  - 1.4.1 Ταξινόμηση ανά Είδος και Δεκαετία . . . 28
  - 1.4.2 Πρόβλεψη Συγχορδιών . . . 29
  - 1.4.3 Αποτελέσματα . . . 31
- 1.5 Συμπεράσματα . . . 35
  - 1.5.1 Συζήτηση . . . 35
  - 1.5.2 Περιορισμοί και Μελλοντική Έρευνα . . . 38

2 Introduction 39

3 Music Theory 41
- 3.1 Pitch and Intervals . . . 42
- 3.2 Scales and Keys . . . 42
- 3.3 Chord, harmony and Chord Progressions . . . 43
- 3.4 Symbolic Representations for Computational Modeling . . . 44
- 3.5 Applications to Music Information Retrieval (MIR) Tasks . . . 44
  - 3.5.1 Genre and Decade Classification . . . 44
  - 3.5.2 Chord Prediction . . . 45
  - 3.5.3 Other tasks . . . 45

4 Machine Learning 47
- 4.1 Learning Paradigms . . . 49
  - 4.1.1 Supervised Learning . . . 49
  - 4.1.2 Unsupervised Learning . . . 49
  - 4.1.3 Reinforcement Learning . . . 49
  - 4.1.4 Semi-Supervised Learning . . . 49

4.1.5 Self-Supervised Learning . . . 49
4.2 Neural Network Fundamentals . . . 50
4.2.1 Perceptrons and Multilayer Networks . . . 50
4.2.2 Activation Functions . . . 51
4.2.3 Loss, Optimization and Backpropagation . . . 51
4.3 Training and Generalization . . . 52
4.4 Deep Learning . . . 53
4.5 Tokenization, Embeddings and Feature Spaces . . . 54

5 Deep Learning for Sequential Data **57**
5.1 Recurrent Neural Networks . . . 59
5.1.1 Vanilla Recurrent Neural Networks . . . 59
5.1.2 Long Short-Term Memory Networks (LSTMs) . . . 60
5.1.3 Gated Recurrent Units (GRUs) . . . 61
5.2 Transformer-based models . . . 62
5.2.1 Positional Embeddings . . . 62
5.2.2 Self-Attention Mechanism . . . 62
5.2.3 Multi-Head Attention . . . 63
5.2.4 Encoder-Decoder Architecture . . . 63
5.2.5 Position-Wise Feed-Forward Network . . . 64
5.2.6 Limitations . . . 64
5.3 State-Space Models and Mamba . . . 64
5.3.1 Foundation State-Space Models . . . 64
5.3.2 Modern Adaptation for Sequence Modeling . . . 65
5.3.3 The Mamba architecture . . . 66
5.4 Architecture Comparison . . . 66
5.5 Related Work . . . 67

6 Latent Distance Models **69**
6.1 Preliminaries . . . 71
6.1.1 Logistic Regression . . . 71
6.1.2 Maximum Likelihood Estimation (MLE) . . . 71
6.1.3 Bayesian Methods . . . 71
6.1.4 Markov Chain Monte Carlo (MCMC) . . . 71
6.2 The Latent Distance Model (LDM) . . . 72
6.3 Variants and Extensions . . . 73
6.4 Limitations . . . 73

7 Proposal **75**
7.1 Motivation and Research Objectives . . . 75
7.2 Overview of Proposed Framework . . . 76

8 Dataset and Analysis **79**
8.1 Dataset Overview . . . 80
8.2 Feature Extraction . . . 80
8.3 Descriptive analysis . . . 81
8.3.1 Structural Features . . . 81
8.3.2 Sequential Features . . . 87
8.4 LDM analysis . . . 90
8.4.1 Dataset Balancing . . . 90
8.4.2 Chord Processing and N-gram Extraction . . . 90
8.4.3 Category Graph . . . 91
8.4.4 N-gram Graph . . . 91
8.4.5 Node Coloring and mapping . . . 91
8.4.6 Latent Distance Modeling . . . 92

9 Experiments **95**
9.1 Genre and Decade Classification . . . 95
9.1.1 Data Representation and Preprocessing . . . 95
9.1.2 Model Architectures . . . 95
9.1.3 Training . . . 95
9.1.4 Evaluation . . . 96
9.2 Chord Prediction . . . 96
9.2.1 Data Representation and Preprocessing . . . 96
9.2.2 Model Architectures . . . 97
9.2.3 Training . . . 97
9.2.4 Evaluation . . . 97
9.3 Results . . . 97
9.3.1 Genre and Decade Classification . . . 97
9.3.2 Chord Prediction . . . 98

10 Conclusion **103**
10.1 Discussion . . . 103
10.2 Limitations and Future Work . . . 105

11 Bibliography **107**

List of Figures

1.1.1 Κύκλος των πέμπτων, σημαντικός για την κατανόηση των σχέσεων μεταξύ των κλειδιών [2]. . . 4
1.1.2 Αναπαράσταση μοντέλου λανθάνουσας απόστασης (α) με χρήση εκτίμησης μέγιστης πιθανοφάνειας, (β) με χρήση μεθόδων [7]. . . 10
1.2.1 Οι δύο ροές επεξεργασίας για τα μοντέλα πρόβλεψης συγχορδιών, (a) η προσέγγιση ενός συμβόλου ανά συγχορδία και (b) η προσέγγιση ενός συμβόλου ανά συστατικό της συγχορδίας (3-όροι). 12
1.3.1 Κατανομή τραγουδιών και καλλιτεχνών ανά είδος και μέσος αριθμός τραγουδιών ανά καλλιτέχνη. 15
1.3.2 Κατανομή τραγουδιών και καλλιτεχνών ανά δεκαετία. . . 16
1.3.3 Κατανομή των ειδών ανά δεκαετία. . . 17
1.3.4 Αριθμός τμημάτων ανά τραγούδι συνολικά, ανά είδος και ανά δεκαετία. . . 17
1.3.5 Μέγεθος τμημάτων συνολικά, ανά είδος και ανά δεκαετία. . . 18
1.3.6 Σύγκριση στατιστικών στοιχείων για τη δομή των χορδών: (α) κατανομή μεγέθους τμημάτων και (β) μέσο μήκος ανά τύπο τμήματος. . . 18
1.3.7 Μέση χρήση μοναδικών συγχορδιών ανά είδος και δεκαετία. . . 19
1.3.8 Κατανομή χρήσης συγχορδιών [14]. . . 19
1.3.9 Κατανομή (α) όλων των ποιοτήτων και (β) των σπάνιων ποιοτήτων, ανά είδος και δεκαετία. . . 20
1.3.10 Κατανομή (α) των επιλογών αντιστροφής και (β) των ποσοστών ύπαρξης αντιστροφής, ανά είδος και δεκαετία. . . 21
1.3.11 Κατανομή των επεκτάσεων ανά είδος και δεκαετία. . . 21
1.3.12 Κινήσεις ρίζας μετρημένες σε ημιτόνια στα διάφορα είδη και δεκαετίες. . . 22
1.3.13 Κορυφαία δυάδες (αριστερά) και τριάδες (δεξιά) σε όλες τις μεταβάσεις και στις μεταβάσεις στην αλλαγή τμημάτων. . . 22
1.3.14 Οι 10 κορυφαίες συγχορδίες και πώς αλλάζουν οι κατανομές των επόμενων συγχορδιών τους κατά τη μετάβαση σε επόμενο μέρος. . . 23
1.3.15 Οι 10 κορυφαίες δυάδες συγχορδιών και πώς αλλάζουν οι κατανομές των επόμενων συγχορδιών τους κατά τη μετάβαση σε επόμενο μέρος. . . 24
1.3.16 Ένα παράδειγμα κόμβων κατηγοριών μετά την πρώτη φάση εκπαίδευσης. . . 26
1.3.17 Ένα παράδειγμα κόμβων πλειάδων μετά τη δεύτερη φάση εκπαίδευσης. . . 27
1.3.18 Ένα παράδειγμα κόμβων κατηγοριών μετά την τρίτη φάση εκπαίδευσης. . . 27
1.3.19 Οι λανθάνουσες θέσεις των τετράδων, πεντάδων και εξάδων, από αριστερά προς τα δεξιά, με βάση τις συνδέσεις που δημιουργήθηκαν από την ομοιότητα των ειδών. . . 28
1.3.20 Οι λανθάνουσες θέσεις των τετράδων, πεντάδων και εξάδων, από αριστερά προς τα δεξιά, με βάση τις συνδέσεις που δημιουργήθηκαν από την ομοιότητα των δεκαετιών. . . 28
1.4.1 Χάρτες θερμότητας της σύγχυσης για το πρόβλημα της ταξινόμησης ανά είδος (αριστερά) και το πρόβλημα ταξινόμησης ανά δεκαετία (δεξιά). . . 32
1.4.2 Σωρευτική Ακρίβεια και Απώλεια για το Τμηματοποιημένο σύνολο δεδομένων. . . 33
1.4.3 Σωρευτική Ακρίβεια, Σωρευτική Απώλεια και Κατανομή Μήκους Ακολουθιών για το Τμηματοποιημένο σύνολο δεδομένων. . . 33
1.4.4 Σωρευτική Ακρίβεια, Σωρευτική Απώλεια και Κατανομή Μήκους Ακολουθιών για το σύνολο δεδομένων Με Ετικέτες. . . 34
1.4.5 Σωρευτική Ακρίβεια, Σωρευτική Απώλεια και Κατανομή Μήκους Ακολουθιών για το Φιλτραρισμένο σύνολο δεδομένων. . . 34

3.2.1 Circle of Fifths, important for understanding relationships between key signatures [2]. . . 43

4.2.1 A single Neuron (Perceptron) [24]. . . . 50
4.2.2 Linear, sigmoid and hyperbolic tangent activation functions. . . . 51
4.2.3 A single Neuron (Perceptron). . . . 51
4.4.1 Multi-Layer Perceptron, 2 hidden layers, 4 units each. . . . 53
4.5.1 Principal Component Analysis in two dimensions, using two principal components [28]. . . . 54
4.5.2 Principal Component Analysis in three dimensions, the colored arrows represent the new axis of the principal components [28]. . . . 55
4.5.3 Hierarchical Clustering Dendogram. . . . 55
4.5.4 The embedding learning process [29]. . . . 56

5.1.1 A recurrent neural unit and its unfold version [30]. . . . 59
5.1.2 A Long Short-Term Memory unit [31]. . . . 61
5.1.3 A Gated Recurrent Unit [36]. . . . 61
5.2.1 The transformer model architecture [37]. . . . 62
5.2.2 Multi-Head Attention: Several attention layers working in parallel [37]. . . . 63
5.3.1 Representation of discrete time state-space model equations. . . . 65
5.3.2 The mamba model architecture including selective memory allocation for efficient computing [39]. . . . 66

6.2.1 Latent Distance Model representation (a) using maximum likelihood estimation, (b) using Bayesian methods [7]. . . . 72

7.2.1 The two pipelines for the chord prediction models, (a) the one token per chord approach and (b) the one token per chord component approach (3-tokens). . . . 77

8.3.1 Song and Artist Distribution by Genre and Average Song number per Artist. . . . 82
8.3.2 Distribution of songs and artists over the decades. . . . 83
8.3.3 Distribution of genres across decades. . . . 84
8.3.4 Number of parts per song, overall, per genre and per decade. . . . 84
8.3.5 Part sizes overall, per genre and per decade. . . . 85
8.3.6 Comparison of chord structure statistics: (a) part-size distribution and (b) average length by section type. . . . 85
8.3.7 Average usage of unique chord per genre and decade. . . . 85
8.3.8 Distribution of chord usage [14]. . . . 86
8.3.9 Distribution of (a) all qualities and (b) outlier qialities, across genres and decades. . . . 86
8.3.10 Distribution of (a) inversion choices and (b) inversion existence percentages, across genres and decades. . . . 87
8.3.11 Distributions of extensions across genres and decades. . . . 88
8.3.12 Root motions measured by semitones across genres and decades. . . . 88
8.3.13 Top bigrams (left) and trigrams (right) over all and on part transitions. . . . 89
8.3.14 Top 10 chords and how the next chord distributions change on part transition. . . . 89
8.3.15 Top 10 bigrams and how the next chord distributions change on part transition. . . . 90
8.4.1 An example of category nodes after the first training phase. . . . 92
8.4.2 An example of n-gram nodes after the second training phase. . . . 93
8.4.3 An example of category nodes after the third training phase. . . . 93
8.4.4 The latent positions of 4-grams, 5-grams and 6-grams, from left to right, based on connections created by genre similarity. . . . 94
8.4.5 The latent positions of 4-grams, 5-grams and 6-grams, from left to right, based on connections created by decade similarity. . . . 94

9.3.1 Confusion heatmaps for the genre classification task (left) and the decade classification task (right). . . . 98
9.3.2 Cumulative Accuracy and Loss for the Standard dataset. . . . 99
9.3.3 Cumulative Accuracy, Cumulative Loss, and Sequence Length Distribution for the Segmented dataset. . . . 100

9.3.4 Cumulative Accuracy, Cumulative Loss, and Sequence Length Distribution for the Labeled dataset. 100
9.3.5 Cumulative Accuracy, Cumulative Loss, and Sequence Length Distribution for the Filtered dataset. 100

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

Η μουσική είναι μια από τις πιο εκφραστικές μορφές της ανθρώπινης δημιουργικότητας, η δομή και οι εσωτερικοί κανόνες της έχουν τραβήξει την προσοχή συνθετών, επιστημόνων και τεχνολόγων για αιώνες. Η κατανόηση του τρόπου με τον οποίο η αρμονία, η μελωδία και ο ρυθμός αλληλεπιδρούν για να δημιουργήσουν συναισθήματα, να εκφράσουν ιδέες και να διηγηθούν ιστορίες, αποτελεί από καιρό σημαντικό στόχο στους τομείς της τέχνης και της επιστήμης. Με την ταχεία επέκταση των ψηφιακών μουσικών δεδομένων και την εξέλιξη των υπολογιστικών εργαλείων, η μελέτη της μουσικής δομής έχει μεταβεί όλο και περισσότερο από την καθαρά θεωρητική ανάλυση σε προσεγγίσεις βασισμένες σε δεδομένα. Το αυξανόμενο ενδιαφέρον και η βελτίωση των μηχανισμών μοντελοποίησης έχουν δώσει τη θέση τους στον τομέα της Ανάκτησης Μουσικής Πληροφορίας (ΑΜΠ), που στοχεύει στην ανάλυση, μοντελοποίηση και κατανόηση της μουσικής, χρησιμοποιώντας υπολογιστικές και στατιστικές μεθόδους.

Στον πυρήνα των μουσικών ακολουθιών βρίσκονται οι συγχορδίες και οι διαδοχές τους. Αποτελούν τα πιο θεμελιώδη στοιχεία για τη μελέτη των τονικών και στυλιστικών χαρακτηριστικών. Αυτές οι διαδοχές καθορίζουν την αρμονική κίνηση, δημιουργώντας τα μοτίβα έντασης και επίλυσης που οργανώνουν τη μουσική μορφή και συμβάλλουν στον χαρακτήρα της. Με τη μοντελοποίησή τους, καθίσταται δυνατή η παρακολούθηση της ιστορικής εξέλιξης της αρμονικής γλώσσας, η αποκάλυψη στυλιστικών χαρακτηριστικών σε διαφορετικά είδη και η εκμάθηση της πρόβλεψης μουσικών γεγονοτών, όπως ποιες συγχορδίες ή διαμορφώσεις μπορεί να ακολουθήσουν.

Ωστόσο, ενώ οι διαδοχές συγχορδιών καταγράφουν βασικές αρμονικές πληροφορίες, αντιπροσωπεύουν μόνο μία διάσταση της μουσικής δομής. Οι συμβολικές ακολουθίες συγχορδιών απεικονίζουν καθαρά αρμονικές αφαιρέσεις, δεν κωδικοποιούν πλήρως τη μελωδική κίνηση, τη ρυθμική δομή, το τέμπο ή τον εκφραστικό συγχρονισμό. Η απόλυτη εστίαση στην τονική ραχοκοκαλιά της μουσικής καθιστά τις συμβολικές ακολουθίες συγχορδιών έναν συνοπτικό αλλά περιορισμένο τρόπο περιγραφής της μουσικής, συμπεριλαμβανομένης της πλούσιας αρμονικής αναπαράστασης.

Οι παραδοσιακές προσεγγίσεις της αρμονικής ανάλυσης βασίζονταν περισσότερο σε συμβολικές μεθόδους βασισμένες σε κανόνες ή σε επιφανειακά στατιστικά μοντέλα που αποτυπώνουν μόνο τις τοπικές σχέσεις μεταξύ των συγχορδιών. Η πρόσφατη εμφάνιση μεθόδων μηχανικής μάθησης και βαθιάς μάθησης έχει δημιουργήσει ένα νέο μονοπάτι προς την κατανόηση της μουσικής. Νευρωνικές αρχιτεκτονικές κατάλληλες για μοντελοποίηση ακολουθιών, όπως τα αναδρομικά νευρωνικά δίκτυα και τις παραλλαγές τους, μοντέλα βασισμένα σε Transformers και, πιο πρόσφατα, μοντέλα κατάστασης-χώρου όπως το Mamba, έχουν εισαγάγει την ικανότητα που απαιτείται για την καταγραφή μακροπρόθεσμων εξαρτήσεων και ιεραρχικών δομών σε μουσικές ακολουθίες. Αυτές οι μέθοδοι επιτρέπουν την εκμάθηση αρμονικών και στυλιστικών σχέσεων, που εξάγονται από ακατέργαστα δεδομένα, χωρίς την ανάγκη ανθρώπινης παρέμβασης. Ο στόχος αυτής της μελέτης είναι η μοντελοποίηση και ανάλυση της μουσικής δομής μέσω συμβολικών ακολουθιών συγχορδιών, χρησιμοποιώντας τόσο στατιστικές μεθόδους όσο και μεθόδους μηχανικής μάθησης, γεφυρώνοντας το χάσμα μεταξύ της μουσικής θεωρίας και της υπολογιστικής μοντελοποίησης και κωδικοποιώντας τις αρμονικές και χρονικές ιδιότητες της μουσικής.

Ενσωματώνοντας στα θεωρητικά θεμέλια της μουσικής, τις σύγχρονες τεχνικές μηχανικής μάθησης, η εργασία αυτή συμβάλλει σε μια βαθύτερη κατανόηση του τρόπου με τον οποίο τα μοντέλα που βασίζονται σε δεδομένα μπορούν να καταγράψουν την ουσία της αρμονικής οργάνωσης. Επιπλέον, παρέχει πληροφορίες για τον τρόπο με τον οποίο οι υπολογιστικές αναπαραστάσεις της μουσικής ευθυγραμμίζονται με την ανθρώπινη αντίληψη και θεωρία, προσφέροντας προοπτικές για μελλοντικές εφαρμογές στην αυτόματη σύνθεση, τη σύσταση μουσικής και τη στυλιστική ανάλυση.

1.1 Θεωρητικό Υπόβαθρο

1.1.1 Μουσική Θεωρία

Η μουσική θεωρία προσφέρει τα εννοιολογικά θεμέλια για την κατανόηση του τρόπου με τον οποίο τα μουσικά στοιχεία, όπως η τονικότητα, η μελωδία, ο ρυθμός, η αρμονία και το ηχόχρωμα, οργανώνουν τον ήχο σε εκφραστικές και σημαντικές δομές, και αυτά τα θεμέλια είναι απαραίτητα για την Ανάκτηση Μουσικών Πληροφοριών (ΑΜΠ).

Στο πεδίο αυτό εφαρμόζονται υπολογιστικές και βασισμένες σε δεδομένα μέθοδοι για την ανάλυση, ερμηνεία και οργάνωση της μουσικής, αξιοποιώντας σε μεγάλο βαθμό συμβολικές αναπαραστάσεις όπως νότες, συγχορδίες και ρυθμικά μοτίβα, που επιτρέπουν στους αλγόριθμους να συλλάβουν θεωρητικές σχέσεις που δεν μπορούν να

εκφραστούν μόνο με τον ακατέργαστο ήχο. Εργασίες όπως η εξαγωγή μελωδίας, η αναγνώριση συγχορδιών, η ανίχνευση κλειδιών, η ταξινόμηση μουσικών ειδών και η ανάλυση ομοιοτήτων βασίζονται στη γνώση της αρμονικής λειτουργίας, του μελωδικού περιγράμματος και της τονικής ιεραρχίας για τη μοντελοποίηση μοτίβων που ευθυγραμμίζονται με την ανθρώπινη μουσική αντίληψη.

Η ενσωμάτωση μουσικοθεωρητικών αρχών στα συστήματα MIR ενισχύει τις αναλυτικές τους δυνατότητες, επιτρέποντας στα μοντέλα να προχωρήσουν πέρα από τις επιφανειακές στατιστικές συσχετίσεις προς μουσικά σημαντικές ερμηνείες. Αυτή η ενότητα θεμελιώνει τις θεωρητικές βάσεις [1] για τα προβλήματα της ανάκτησης μουσικής πληροφορίας που εξετάζονται στη διπλωματική.

Τόνος και διαστήματα

Η τονικότητα είναι το αντιληπτό ύψος ή βάθος ενός ήχου, που καθορίζεται από τη συχνότητά του και στη δυτική μουσική, οι τονικότητες που απέχουν μια οκτάβα (αναλογία συχνότητας 2:1) θεωρούνται ισοδύναμες και ομαδοποιούνται σε δώδεκα διακριτές κατηγορίες τονικότητας.

Ένα διάστημα είναι η απόσταση μεταξύ δύο τονικοτήτων που μετράται σε ημιτόνια, και περιγράφει τόσο ποσοτικές όσο και ποιοτικές σχέσεις (για παράδειγμα μια μείζονα τρίτη εκτείνεται σε τέσσερα ημιτόνια και μια τέλεια πέμπτη σε επτά ημιτόνια μακριά από τη ρίζα). Στην ανάκτηση μουσικών πληροφοριών, οι τόνοι και τα διαστήματα κωδικοποιούνται συνήθως συμβολικά, είτε ως σύνολα κατηγοριών τόνου είτε ως ακολουθίες διαστημάτων, επιτρέποντας στα μοντέλα να καταγράφουν τη μελωδική και αρμονική δομή, παρέχοντας ταυτόχρονα αμεταβλητότητα στη μετατόπιση. Αυτές οι αναπαραστάσεις απλοποιούν τα μουσικά δεδομένα βοηθώντας στην ευθυγράμμιση με τις θεωρητικές περιγραφές της μελωδίας και της αρμονίας, σχηματίζοντας τη βάση για την ανάλυση πιο σύνθετων δομών, όπως κλίμακες, κλειδιά, συγχορδίες και προόδους.

Κλίμακες και Κλειδιά

Οι κλίμακες είναι οργανωμένες ακολουθίες τόνων που αποτελούν τη βάση της μελωδίας και της αρμονίας, με τις μείζονες και ελάσσονες κλίμακες να είναι οι πιο συνηθισμένες στη δυτική μουσική, οι οποίες ορίζονται από συγκεκριμένα μοτίβα ολόκληρων και μισών βημάτων σε ημιτόνια.

Ένα κλειδί ορίζει το τονικό κέντρο ενός κομματιού, συνήθως την τονική νότα μιας κλίμακας, καθιερώνοντας μια ιεραρχία μεταξύ τόνων και συγχορδιών που αναθέτει ρόλους όπως τονική, κυρίαρχη, υποκυρίαρχη, μεσαία και άλλους, οι οποίοι καθοδηγούν την αρμονική λειτουργία και την πρόοδο. Ο κύκλος των πέμπτων παρέχει μια συστηματική απεικόνιση των σχέσεων μεταξύ των κλειδιών, δείχνοντας ποια κλειδιά μοιράζονται τόνους και πώς οι μετατροπές σε κοντινά κλειδιά ακούγονται φυσικές. Στην Ανάκτηση Μουσικής Πληροφορίας, οι κλίμακες και τα κλειδιά συχνά αναπαρίστανται συμβολικά, ως σύνολα τάξεων τόνων ή ακολουθίες βαθμών κλίμακας σε σχέση με την τονική, επιτρέποντας στα υπολογιστικά μοντέλα να αναλύουν μελωδίες και αρμονικές δομές με τρόπο ανεξάρτητο από το κλειδί, διατηρώντας παράλληλα το τονικό πλαίσιο.

Συγχορδίες και Διαδοχές Συγχορδιών

Οι συγχορδίες είναι σύνολα τόνων που ηχούν ταυτόχρονα, με τις τριάδες, την απλούστερη μορφή, να αποτελούνται από μια ρίζα, μια τρίτη και μια πέμπτη, και μπορούν να επεκταθούν με επιπλέον τόνους όπως έβδομες ή ένατες.

Η ποιότητα της συγχορδίας, που ορίζεται από τα διαστήματα μεταξύ της ρίζας και των άλλων νοτών, τις ταξινομεί σε τύπους όπως μείζονα, ελάσσονα, αυξημένη ή μειωμένη, ενώ οι αναστροφές αναδιατάσσουν τη νότα του μπάσου (τη χαμηλότερη συχνοτικά νότα) της συγχορδίας χωρίς να αλλάζουν το περιεχόμενο των τόνων της, επιτρέποντας μια ομαλότερη αρμονική κίνηση. Η αρμονία προκύπτει από τις αλληλεπιδράσεις μεταξύ των νοτών των συγχορδιών και μεταξύ των ίδιων των συγχορδιών με την πάροδο του χρόνου, δημιουργώντας ένταση και επίλυση, και δομείται μέσω αρχών όπως η λειτουργική αρμονία και η καθοδήγηση των φωνών, που περιγράφουν τους ρόλους των συγχορδιών και τη σταδιακή κίνηση των μεμονωμένων νοτών για την επίτευξη ομαλών μεταβάσεων.

Οι διαδοχές συγχορδιών είναι ακολουθίες συγχορδιών που καθορίζουν την τονική δομή και τη μουσική κατεύθυνση, με κοινά μοτίβα όπως I-IV-V-I ή ii-V-I να παρέχουν κύκλους έντασης και επίλυσης. Στην ανάκτηση μουσικών πληροφοριών, οι αρμονικές πληροφορίες μπορούν να κωδικοποιηθούν συμβολικά ως ετικέτες συγχορδιών, σύνολα τόνων ή αναπαραστάσεις διαστημάτων, επιτρέποντας σε ακολουθιακά μοντέλα όπως τα

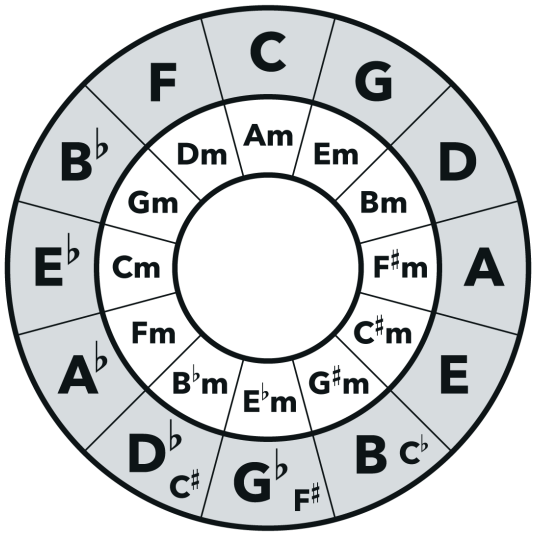


Figure 1.1.1: Κύκλος των πέμπτων, σημαντικός για την κατανόηση των σχέσεων μεταξύ των κλειδιών [2].

αναδρομικά δίκτυα ή οι Transformers να αναλύουν τόσο τις τοπικές όσο και τις συνολικές τονικές σχέσεις για προβλήματα όπως η αναγνώριση και πρόβλεψη συγχορδιών, η ανίχνευση κλειδιών και η ταξινόμηση στυλ.

Συμβολικές Αναπαραστάσεις και Προβλήματα Ανάκτησης Μουσικής Πληροφορίας

Οι συμβολικές αναπαραστάσεις μουσικής κωδικοποιούν χαρακτηριστικά όπως τον τόνο, τη διάρκεια και την αρμονία ως κατηγορικές ή αριθμητικές τιμές, επιτρέποντας στα υπολογιστικά μοντέλα να μαθαίνουν τη μουσική δομή πέρα από τον ακατέργαστο ήχο. Μορφές όπως τα αρχεία MIDI και οι επισημειώσεις συγχορδιών παρέχουν λεπτομερείς πληροφορίες για τις νότες, τις συγχορδίες και το αρμονικό περιεχόμενο, οι οποίες μπορούν να μετατραπούν σε αριθμητικές κωδικοποιήσεις, διανύσματα one-hot, σύνολα τάξεων τόνου ή χαρακτηριστικά χρώματος για μοντελοποίηση.

Οι διαδοχικές και αρμονικές σχέσεις στη μουσική μπορούν να καταγραφούν χρησιμοποιώντας πλειάδες, λανθάνοντες χώρους ή αρχιτεκτονικές όπως αναδρομικά νευρωνικά δίκτυα, Transformers και μοντέλα κατάστασης-χώρου, επιτρέποντας την πρόβλεψη συγχορδιών, την ανάλυση των διαδοχών και τον τονικό χαρακτηρισμό. Αυτές οι αναπαραστάσεις υποστηρίζουν ένα ευρύ φάσμα προβλημάτων ανάκτησης μουσικής πληροφορίας, συμπεριλαμβανομένης της ταξινόμησης ανά είδος και δεκαετία, οι οποίες βασίζονται σε μοτίβα τόνου, συγχορδιών και διαδοχών για τον προσδιορισμό του στυλ ή των χρονικών συμφραζομένων, καθώς και της πρόβλεψης και δημιουργίας συγχορδιών, οι οποίες μοντελοποιούν την αρμονική συνέχεια και τις στυλιστικές τάσεις. Πέρα από αυτά, οι συμβολικές αναπαραστάσεις διευκολύνουν επίσης την ανίχνευση κλειδιών, την παρακολούθηση ρυθμού, την εξαγωγή μελωδίας, την αρμονική ανάλυση, την ομοιότητα μουσικής και την αναγνώριση στυλ, παρέχοντας ένα ισχυρό πλαίσιο για την ανάλυση και τη μοντελοποίηση της μουσικής με έναν υπολογιστικά και μουσικά ουσιαστικό τρόπο.

1.1.2 Ακολουθιακά Μοντέλα

Η μουσική είναι εγγενώς διαδοχική και η δομή της, που βασίζεται σε χρονικά εξαρτώμενα μοτίβα όπως η ένταση, η επίλυση και η επανάληψη, απαιτεί μοντέλα που μπορούν να καταγράψουν τις χρονικές σχέσεις. Η βαθιά μάθηση το κατέστησε αυτό δυνατό με την εισαγωγή αρχιτεκτονικών ικανών να μαθαίνουν μακροπρόθεσμες εξαρτήσεις, ξεκινώντας από τα αναδρομικά νευρωνικά δίκτυα και τις παραλλαγές τους με πύλες, τα οποία χρησιμοποιούν μηχανισμούς μνήμης για να διατηρούν τα συμφραζόμενα. Πιο πρόσφατες προσεγγίσεις, συμπεριλαμβανομένων των μοντέλων προσοχής που βασίζονται σε Transformers και των μοντέλων κατάστασης-χώρου όπως το Mamba, βελτιώνουν τη μακροπρόθεσμη μοντελοποίηση και την παράλληλη επεξεργασία, καθιστώντας τα αποτελεσματικά για εκτεταμένες μουσικές ακολουθίες και πολυτροπικά δεδομένα. Αυτή η υποενότητα περιγράφει αυτές τις

αρχιτεκτονικές, τους μηχανισμούς τους και τον ρόλο τους στην ανάκτηση μουσικών πληροφοριών, εξηγώντας πώς μαθαίνουν και δημιουργούν μουσικά σημαντική δομή.

Αναδρομικά Νευρωνικά Δίκτυα

Τα παραδοσιακά δίκτυα προώθησης αγνοούν τις χρονικές εξαρτήσεις. Οι πρώτες προσπάθειες αντιμετώπιζαν τον χρόνο ως μια επιπλέον μεταβλητή χρησιμοποιώντας χωρικές αναπαραστάσεις, αλλά απαιτούσαν σταθερό μήκος εισόδων, δεν μπορούσαν να καθορίσουν πόσες προσωρινές αποθηκεύσεις χρειάζονταν ή πότε έπρεπε να εξεταστεί κάθε δείγμα και δεν μπορούσαν να διακρίνουν την απόλυτη από τη σχετική χρονική θέση. Αντί να αντιμετωπίζεται ο χρόνος ως μια νέα διάσταση, αναπτύχθηκε μια απλή αρχιτεκτονική αναδρομικού νευρωνικού δικτύου που αξιοποιεί την επίδραση του χρόνου στα δεδομένα χρησιμοποιώντας το χρονικό πλαίσιο ως είσοδο [3]. Η κρυφή του κατάσταση μεταφέρει πληροφορίες σε όλα τα χρονικά βήματα, επιτρέποντας στο δίκτυο να θυμάται προηγούμενες εισόδους και να παράγει μελλοντικές εξόδους, ώστε τα αναδρομικά νευρωνικά δίκτυα να καταγράφουν τόσο τις βραχυπρόθεσμες όσο και τις μακροπρόθεσμες εξαρτήσεις για εργασίες διαδοχικής ανάλυσης.

Απλά Αναδρομικά Νευρωνικά Δίκτυα Τα απλά αναδρομικά νευρωνικά δίκτυα αποτελούνται από μια κρυφή κατάσταση h_t που ενημερώνεται σε κάθε χρονικό βήμα με βάση την προηγούμενη κρυφή κατάσταση h_{t-1} και την είσοδο x_t:

$$h_t = \phi\big(W_x x_t + W_h h_{t-1} + b_t\big) \tag{1.1.1}$$

Όπου ϕ αντιπροσωπεύει τη μη γραμμική συνάρτηση ενεργοποίησης, W_x και W_h είναι οι πίνακες βαρών της εισόδου του επιπέδου και της προηγούμενης κρυφής κατάστασης του επιπέδου $h_{(t-1)}$ αντίστοιχα και b_t αντιστοιχεί στον διάνυσμα μεροληψίας.

Τα στοιβαγμένα αναδρομικά νευρωνικά δίκτυα διαδίδουν την κρυφή κατάσταση ενός κατώτερου επιπέδου ως είσοδο στο επόμενο επίπεδο, επιτρέποντας στα ανώτερα επίπεδα να μάθουν πιο αφηρημένες χρονικές αναπαραστάσεις.

Η εκπαίδευση με οπισθοδιάδοση μέσω του χρόνου (Backpropagation Through Time) μπορεί να υποστεί από εξαφανιζόμενες ή εκρηκτικές κλίσεις (gradients):

$$\frac{\partial L}{\partial h_l} = \frac{\partial L}{\partial h_L} \prod_{n=l+1}^{L} \frac{\partial h_n}{\partial h_{n-1}} \tag{1.1.2}$$

Εάν η νόρμα κάθε όρου $\frac{\partial h_n}{\partial h_{n-1}}$ είναι <1, η κλίση εξαφανίζεται. Εάν είναι >1, εκρήγνυται, προκαλώντας και στις δύο περιπτώσεις αναποτελεσματική εκπαίδευση σε βαθιά δίκτυα.

Αυτή η αστάθεια της κλίσης περιορίζει τα απλά αναδρομικά νευρωνικά δίκτυα να μοντελοποιούν αποτελεσματικά σχετικά σύντομες ακολουθίες [4].

Δίκτυα Μακράς και Βραχείας Μνήμης Τα δίκτυα μακράς και βραχείας μνήμης [5] επεκτείνουν τα απλά αναδρομικά νευρωνικά δίκτυα για να χειρίζονται μακρές ακολουθίες ρυθμίζοντας τη ροή πληροφοριών μέσω ενός κυττάρου μνήμης. Χωρίζουν τις χρονικές πληροφορίες σε ένα βραχυπρόθεσμο μέρος (κρυφή κατάσταση h_t) και ένα μακροπρόθεσμο μέρος (κατάσταση κελιού c_t), μετριάζοντας το πρόβλημα της εξαφάνισης της κλίσης. Κάθε κελί μακράς και βραχείας μνήμης χρησιμοποιεί τρεις πύλες: λήθης f_t, εισόδου i_t και εξόδου o_t, οι οποίες ελέγχουν τη διατήρηση και την απορρόφηση νέων πληροφοριών αλλά και την έκθεση της κατάστασης του κελιού, αντίστοιχα:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f\right), \tag{1.1.3}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i\right), \tag{1.1.4}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o\right), \tag{1.1.5}$$

$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c\right), \tag{1.1.6}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{1.1.7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right). \tag{1.1.8}$$

Όπου σ δηλώνει τη σιγμοειδή συνάρτηση ενεργοποίησης και tanh είναι η υπερβολική εφαπτομενική συνάρτηση ενεργοποίησης, $\odot$ αντιπροσωπεύει τον πολλαπλασιασμό στοιχείων και b_f, b_i, b_o, b_c είναι οι παράμετροι μετατόπισης των πυλών λήθης, εισόδου, εξόδου και κελιού αντίστοιχα.

Αυτές οι πύλες επιτρέπουν στο δίκτυο να διατηρεί μια σχεδόν σταθερή ροή σφαλμάτων, σταθεροποιώντας τη διάδοση της κλίσης σε μεγάλες ακολουθίες. Ενώ τα ίδια υπερτερούν των απλών αναδρομικών νευρωνικών δικτύων σε μεγαλύτερες εξαρτήσεις, εξακολουθούν να απαιτούν προσεκτική ρύθμιση των υπερπαραμέτρων και συνεπάγονται υψηλότερο κόστος μνήμης και υπολογιστικής ισχύος.

Αναδρομικές Μονάδες με Πύλες Οι αναδρομικές μονάδες με πύλες είναι μια απλοποιημένη παραλλαγή των δικτύων μακράς και βραχείας μνήμης, που χρησιμοποιεί λιγότερους παραμέτρους, ενώ εξακολουθεί να καταγράφει μακροπρόθεσμες εξαρτήσεις. Η παραλλαγή αυτή χρησιμοποιεί μια πύλη ενημέρωσης z_t και μια πύλη επαναφοράς r_t για τον έλεγχο της ροής πληροφοριών από την προηγούμενη κρυφή κατάσταση h_{t-1} στην υποψήφια κρυφή κατάσταση $\tilde{h}_t$, ενημερώνοντας την κρυφή κατάσταση h_t ως εξής:.

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z\right), \tag{1.1.9}$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r\right), \tag{1.1.10}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \left(\mathbf{r}_t \odot \mathbf{h}_{t-1}\right) + \mathbf{b}_h\right), \tag{1.1.11}$$

$$\mathbf{h}_t = \left(\mathbf{1} - \mathbf{z}_t\right) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t. \tag{1.1.12}$$

Η πύλη ενημέρωσης z_t εξισορροπεί την επίδραση της προηγούμενης κρυφής κατάστασης και της υποψήφιας κατάστασης, ενώ η πύλη επαναφοράς r_t ελέγχει το βαθμό στον οποίο η προηγούμενη κρυφή κατάσταση επηρεάζει την υποψήφια. Οι αναδρομικές μονάδες με πύλες διατηρούν αποτελεσματικά τις μακροπρόθεσμες εξαρτήσεις με λιγότερους παραμέτρους από τα δίκτυα μακράς και βραχείας μνήμης , εκπαιδεύονται γρηγορότερα και μειώνουν το κόστος μνήμης, αν και εξακολουθούν να αντιμετωπίζουν δυσκολίες με εξαιρετικά μεγάλες ακολουθίες και δεν μπορούν να παραλληλοποιηθούν σε χρονικά βήματα [6].

Μοντέλα Βασισμένα σε Transformer

Οι Transformers εξαλείφουν την αναδρομικότητα, αντιμετωπίζοντας το χρόνο ως μια επιπλέον διάσταση, και βασίζονται σε ενσωματώσεις θέσης και μηχανισμούς αυτοπροσοχής (self-attention) για να καταγράψουν παράλληλα τις χρονικές εξαρτήσεις. Οι ενσωματώσεις θέσης κωδικοποιούν τη σειρά της ακολουθίας, είτε σταθερή είτε εκμαθημένη. Ο μηχανισμός αυτοπροσοχής υπολογίζει αναπαραστάσεις που λαμβάνουν υπόψη τα συμφραζόμενα:

$$\text{Προσοχή}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{1.1.13}$$

Όπου d_k είναι η διάσταση των διανυσμάτων κλειδιών και χρησιμοποιείται για τη σταθεροποίηση των κλίσεων κατά τη διάρκεια της εκπαίδευσης.

Οι μετασχηματιστές μπορούν επίσης να χρησιμοποιούν την προσοχή πολλαπλών κεφαλών για να καταγράφουν παράλληλα διαφορετικούς τύπους σχέσεων. Η αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή αποτελείται από στοιβαγμένα επίπεδα με προσοχή πολλαπλών κεφαλών και δίκτυα προώθησης κατά θέση (position-wise feed-forward networks), επιτρέποντας την αποτελεσματική μοντελοποίηση ακολουθιών. Το δίκτυο προώθησης ($\Delta\Pi$) είναι:

$$\Delta\Pi(\mathbf{x}) = \phi\big(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\big)\mathbf{W}_2 + \mathbf{b}_2 \tag{1.1.14}$$

Παρά την επιτυχία τους, οι Transformers έχουν τετραγωνική υπολογιστική και μνημονική πολυπλοκότητα $O(n^2)$ ως προς το μήκος της ακολουθίας, ενδέχεται να χάσουν πληροφορίες λόγω έλλειψης αναδρομικότητας και απαιτούν μεγάλα σύνολα δεδομένων και υψηλούς υπολογιστικούς πόρους. Πρόσφατες προσεγγίσεις, όπως η γραμμική προσοχή ή τα υβριδικά μοντέλα κατάστασης-χώρου (όπως Mamba), στοχεύουν στο να συνδυάσουν την αποτελεσματικότητα της προσοχής με τα πλεονεκτήματα της αναδρομικότητας.

Μοντέλα κατάστασης-χώρου και Mamba

Τα μοντέλα κατάστασης-χώρου διατηρούν μια λανθάνουσα κατάσταση $\mathbf{x}_t$ που εξελίσσεται με την πάροδο του χρόνου, καταγράφοντας χρονικές εξαρτήσεις και επηρεάζοντας τα αποτελέσματα $\mathbf{y}_t$. Στη βασική τους μορφή, περιγράφονται ως:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t, \tag{1.1.15}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t. \tag{1.1.16}$$

Η έξοδος μπορεί επίσης να εκφραστεί αναδρομικά:

$$\mathbf{y}_t = \sum_{k=0}^{t} \mathbf{C}\mathbf{A}^{\,t-k}\mathbf{B}\mathbf{u}_k + \mathbf{D}\mathbf{u}_t \tag{1.1.17}$$

Ή ισοδύναμα σε σημειογραφία συνέλιξης χρησιμοπιώντας $\mathbf{K} = \big\{\mathbf{CB},\ \mathbf{CAB},\ \ldots,\ \mathbf{CA}^k\mathbf{B},\ \ldots\big\}$ ως πηρήνα (kernel):

$$\mathbf{y} = \mathbf{K}\mathbf{u} + \mathbf{D}\mathbf{u} \tag{1.1.18}$$

Σύγχρονες παραλλαγές όπως το S4 παραμετροποιούν αποτελεσματικά το $\mathbf{A}$ για σταθερό, παράλληλο υπολογισμό, καταγράφοντας μακροπρόθεσμες εξαρτήσεις με γραμμική πολυπλοκότητα στο μήκος της ακολουθίας, καθιστώντας τις επεκτάσιμες για εργασίες βαθιάς μάθησης σε γλώσσα, ήχο και χρονοσειρές. Οι περιορισμοί περιλαμβάνουν τη δυσκολία μοντελοποίησης εξαιρετικά δυναμικών ακολουθιών, τα εμπόδια στη μνήμη και την εστίαση σε καθολικά μοτίβα έναντι τοπικών.

Η αρχιτεκτονική Mamba Το Mamba επεκτείνει το S4 εισάγοντας επιλεκτικές παραμέτρους χώρου κατάστασης που προσαρμόζονται δυναμικά στην είσοδο, με αποτέλεσμα παρόμοιο με την προσοχή, αλλά με γραμμικό κόστος. Υποστηρίζει επίσης συμπεριφορά πολλαπλών κλιμάκων (multi-scale), καταγράφοντας εξαρτήσεις σε διαφορετικές χρονικές διαστάσεις. Η αρχιτεκτονική του έχει σχεδιαστεί για γραμμική κλιμάκωση ως προς το μήκος της ακολουθίας, σταθερή μνήμη ανά σύμβολο (token) και παραλληλοποίηση με αξιοποίηση του υλικού (hardware). Τα μειονεκτήματα περιλαμβάνουν αυξημένη πολυπλοκότητα, δυσκολότερη ερμηνευσιμότητα και πιθανή αναποτελεσματικότητα για σύντομες ακολουθίες ή εργασίες με περιορισμένα δεδομένα.

Σύγκριση Αρχιτεκτονικών

Τα διαδοχικά μοντέλα έχουν εξελιχθεί ώστε να εξισορροπούν τη μακροπρόθεσμη μοντελοποίηση και την αποδοτικότητα. Τα απλά αναδρομικά νευρωνικά δίκτυα εισάγουν κρυφές καταστάσεις, αλλά υποφέρουν από εξαφανιζόμενες/εκρηκτικές κλίσεις και έλλειψη παραλληλοποίησης, ενώ οι παραλλαγές τους που χρησιμοποιούν

πύλες βελτιώνουν τη σταθερότητα, αλλά εξακολουθούν να αντιμετωπίζουν προβλήματα με πολύ μεγάλες ακολουθίες. Οι Transformers καταργούν την αναδρομικότητα, χρησιμοποιώντας αυτο-προσοχή και πληροφορίες θέσης για να καταγράψουν τις καθολικές εξαρτήσεις, αλλά η τετραγωνική πολυπλοκότητά τους τους καθιστά δαπανηρούς για μακρές εισόδους. Το Mamba συνδυάζει τα πλεονεκτήματα των αναδρομικών και των μοντέλων που βασίζονται στην προσοχή μέσω ενός επιλεκτικού μηχανισμού κατάστασης-χώρου που κλιμακώνεται γραμμικά με το μήκος της ακολουθίας και καταγράφει τόσο τη βραχυπρόθεσμη όσο και τη μακροπρόθεσμη δομή, αν και η πολυπλοκότητά του μπορεί να εμποδίσει την απόδοση σε μικρότερες ακολουθίες όπου απλούστερα αναδρομικά μοντέλα μπορεί να είναι πιο αποτελεσματικά.

1.1.3 Μοντέλα Λανθάνουσας Απόστασης

Τα Μοντέλα Λανθάνουσας Απόστασης (ΜΛΑ) αναπαριστούν δίκτυα ενσωματώνοντας κόμβους σε έναν λανθάνοντα χώρο, συνήθως ευκλείδειο, όπου η απόσταση μεταξύ των κόμβων καθορίζει την πιθανότητα μιας σύνδεσης. Οι κόμβοι που βρίσκονται πιο κοντά σε αυτόν τον χώρο είναι πιο πιθανό να σχηματίσουν ακμές, επιτρέποντας στο μοντέλο να καταγράψει δομικά μοτίβα όπως μεταβατικότητα, ομαδοποίηση και ομοφιλία χωρίς προδιαγραφή χαρακτηριστικών με ανθρώπινη παρέμβαση. Τα μοντέλα αυτά παρέχουν ένα γεωμετρικό και ερμηνεύσιμο πλαίσιο που μπορεί να εφαρμοστεί πέρα από τα κοινωνικά δίκτυα, στα οποία χρησιμοποίηθηκαν αρχικά, για παράδειγμα σε μουσικά στοιχεία όπως συγχορδίες ή μοτίβα, αποκαλύπτοντας κρυφές σχέσεις και δομικά μοτίβα που δεν είναι εμφανή από την επιφανειακή ανάλυση. Αυτό υπογραμμίζει τόσο την ευελιξία των μοντέλων λανθάνουσας απόστασης όσο και τη χρησιμότητά τους ως βάση για την κατανόηση σύνθετων σχεσιακών δεδομένων.

Βασικές Αρχές

Τα μοντέλα λανθάνουσας απόστασης, όπως και πολλά πιθανοτικά μοντέλα, βασίζονται σε στατιστικές μεθόδους για την εκτίμηση παραμέτρων και λανθανουσών μεταβλητών. Οι μέθοδοι που αναφέρονται σε αυτή την υποενότητα είναι οι στατιστικές μέθοδοι που εξηγούνται παρακάτω.

Λογιστικής Παλινδρόμησης Τα μοντέλα λογιστικής παλινδρόμησης είναι στατιστικά μοντέλα για δυαδικά αποτελέσματα. Καθορίζουν τον τρόπο με τον οποίο τα χαρακτηριστικά x επηρεάζουν το αποτέλεσμα ενός γεγονότος y χρησιμοποιώντας τη λογιστική συνάρτηση:

$$\Pr(y = 1 \mid x) = \frac{1}{1 + e^{-x^\top \beta}} \tag{1.1.19}$$

Με β να είναι οι παράμετροι του μοντέλου. Οι λογαριθμικές πιθανότητες του γεγονότος είναι γραμμικές σε σχέση με τα χαρακτηριστικά:

$$\log\left(\frac{\Pr(y = 1 \mid x)}{\Pr(y = 0 \mid x)}\right) = x^\top \beta \tag{1.1.20}$$

Στα μοντέλα λανθάνουσας απόστασης, αυτή η συνάρτηση καθορίζει την πιθανότητα ύπαρξης ακμών με βάση τις αποστάσεις στον λανθάνοντα χώρο.

Εκτίμηση Μέγιστης Πιθανοφάνειας (ΕΜΠ) Η εκτίμηση μέγιστης πιθανοφάνειας είναι μια μέθοδος εκτίμησης παραμέτρων που αποφασίζει με βάση ποιες τιμές παραμέτρων μεγιστοποιείται η πιθανοφάνεια των παρατηρούμενων δεδομένων:

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} L(\theta; \text{δεδομένα}) \tag{1.1.21}$$

Όπου $L(\theta; \text{δεδομένα}) = \Pr(\text{δεδομένα} \mid \theta)$.

Στα μοντέλα λανθάνουσας απόστασης, η εκτίμηση μέγιστης πιθανοφάνειας μπορεί να εκτιμήσει τη λανθάνουσα θέση z_i και παραμέτρους όπως το β που καθιστούν το παρατηρούμενο δίκτυο πιο πιθανό να έχει προκύψει στο πλαίσιο του μοντέλου.

Μπεϋζιανές Μέθοδοι Η μπεϋζιανή συμπερασματολογία αντιμετωπίζει τις παραμέτρους ως τυχαίες μεταβλητές που έχουν προηγούμενες κατανομές $\mathrm{p}(\vartheta)$ και χρησιμοποιεί τα παρατηρούμενα δεδομένα για να σχηματίσει την εκ των υστέρων κατανομή:

$$\Pr(\theta \mid \text{δεδομένα}) \propto \Pr(\text{δεδομένα} \mid \theta)\ \Pr(\theta) \tag{1.1.22}$$

Όπου $\Pr(\text{δεδομένα} \mid \theta)$ είναι η πιθανότητα των δεδομένων με βάση τις παραμέτρους θ.

Σε αντίθεση με την εκτίμηση μέγιστης πιθανότητας, η οποία καθορίζει τις τιμές των παραμέτρων, οι μπεϋζιανές μέθοδοι παρέχουν μια κατανομή των παραμέτρων.

Αλυσίδα Markov Monte Carlo Η αλυσίδα Markov Monte Carlo είναι μια υπολογιστική μέθοδος για τη δειγματοληψία από μια σύνθετη κατανομή πιθανοτήτων, όπως τα μπεϋζιανά μοντέλα. Κατασκευάζει μια ακολουθία εξαρτώμενων δειγμάτων που προσεγγίζουν την κατανομή-στόχο με την πάροδο του χρόνου.

Το Βασικό Μοντέλο

Το Μοντέλο Λανθάνουσας Απόστασης Τα μοντέλα λανθάνουσας απόστασης αναπαριστούν δίκτυα αντιστοιχίζοντας κάθε κόμβο i με το διάνυσμα z_i σε έναν λανθάνοντα ευκλείδειο χώρο διάστασης $d, z_i \in \mathbb{R}^d$. Η ιδέα πίσω από αυτό το μοντέλο είναι ότι η απόσταση που χωρίζει δύο κόμβους αντανακλά την πιθανότητα ύπαρξης μιας ακμής μεταξύ τους. Οι κόμβοι που βρίσκονται πιο κοντά μεταξύ τους είναι πιο πιθανό να μοιράζονται μια σύνδεση από ό,τι εκείνοι που βρίσκονται πιο μακριά.

Για τη μη κατευθυνόμενη έκδοση αυτού του μοντέλου, η πιθανότητα ύπαρξης μιας ακμής μεταξύ των κόμβων i και j μοντελοποιείται χρησιμοποιώντας μια λογιστική συνάρτηση σύνδεσης [7]:

$$\Pr(Y_{i,j} = 1 \mid z_i, z_j, \beta) = \frac{1}{1 + e^{\|z_i - z_j\| - \beta}} \tag{1.1.23}$$

Ή σε μορφή logit:

$$\text{logit}\ \Pr(Y_{i,j} = 1 \mid z_i, z_j, \beta) = \beta - \|z_i - z_j\| \tag{1.1.24}$$

Στους παραπάνω τύπους , $Y_{i,j}$ είναι ο πίνακας γειτνίασης, $Y_{i,j} = 1$ υποδηλώνει την ύπαρξη ακμής και $Y_{i,j} = 0$ το αντίθετο, $\|z_i - z_j\|$ υποδηλώνει την ευκλείδεια απόσταση μεταξύ των κόμβων i και j και β είναι η καθολική παράμετρος που ελέγχει την πυκνότητα του λανθάνοντος χώρου. Μικρότερες τιμές β μειώνουν την πιθανότητα μιας ακμής για μια δεδομένη απόσταση, ενώ μεγαλύτερες τιμές την αυξάνουν, μετατοπίζοντας αποτελεσματικά τη λογιστική καμπύλη και περιορίζοντας έμμεσα τη γεωμετρία, έτσι ώστε το μοντέλο να μπορεί να προσαρμοστεί στα δεδομένα.

Η εγγύτητα στον λανθάνοντα χώρο υποδηλώνει εννοιολογικά κόμβους που είναι λειτουργικά παρόμοιοι. Αυτή η δομή επιτρέπει στο μοντέλο να εκφράζει και να καταγράφει ιδιότητες του δικτύου, όπως η μεταβατικότητα, η ομαδοποίηση και η ομοφιλία, χωρίς να απαιτείται ρητή κωδικοποίηση χαρακτηριστικών. Τα κατευθυνόμενα δίκτυα μπορούν επίσης να υλοποιηθούν με την εκχώρηση ξεχωριστών λανθανόντων διανυσμάτων για τους κόμβους αποστολής και λήψης, τροποποιώντας τον όρο απόστασης για να προσαρμοστεί στην κατεύθυνση της ακμής.

Οι παράμετροι του μοντέλου, οι λανθάνουσες θέσεις z_i και β, προκύπτουν από τον πίνακα γειτνίασης που παρατηρήθηκε χρησιμοποιώντας μεθόδους στατιστικής συμπερασματολογίας. Η εκτίμηση μέγιστης πιθανοφάνειας δίνει μια ενιαία διαμόρφωση που μεγιστοποιεί την πιθανοφάνειας του δικτύου που παρατηρήθηκε, ενώ η μπεϋζιανή συμπερασματολογία, σε συνδυασμό με τη δειγματοληψία αλυσίδας Markov Monte Carlo, δημιουργεί μια κατανομή στις λανθάνουσες θέσεις και παραμέτρους, εισάγοντας αβεβαιότητα στο μοντέλο και καθιστώντας το πιο ευέλικτο.

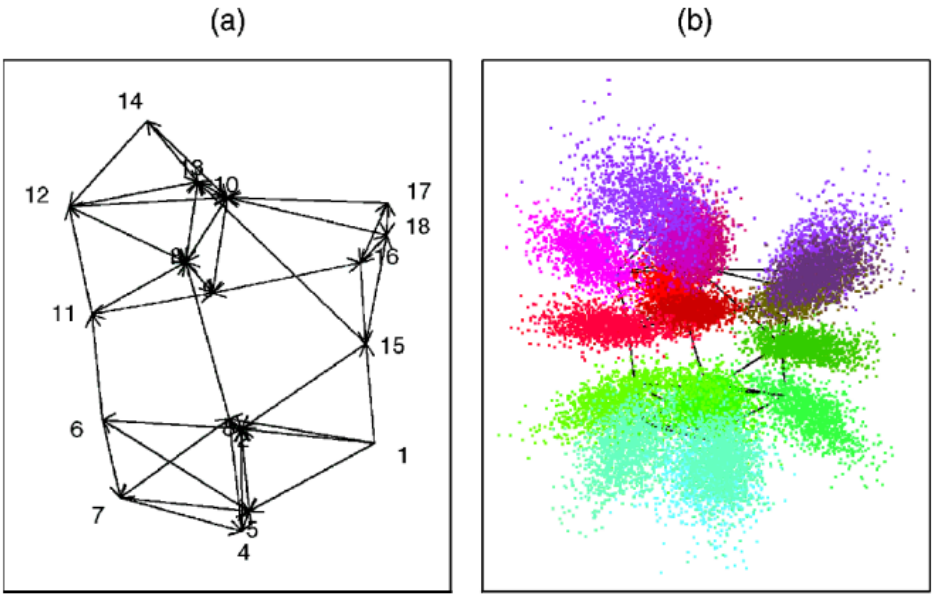


Figure 1.1.2: Αναπαράσταση μοντέλου λανθάνουσας απόστασης (α) με χρήση εκτίμησης μέγιστης πιθανοφάνειας, (β) με χρήση μεθόδων [7].

Περιορισμοί και Παραλλαγές

Ενώ το βασικό μοντέλο λανθάνουσας απόστασης προσφέρει ένα απλό και ερμηνεύσιμο πλαίσιο για σχεσιακά δεδομένα, υπάρχουν αρκετές επεκτάσεις, συμπεριλαμβανομένων μοντέλων που καταγράφουν τη δομή της κοινότητας, τη χρονική δυναμική, πρόσθετες συνδιακύμαντες μεταβλητές και εναλλακτικές γεωμετρίες, όπως υπερβολικούς ή σφαιρικούς χώρους, για την καλύτερη αναπαράσταση ιεραρχικών δικτύων ή δικτύων χωρίς κλίμακα (scale-free) [8], [9], [10], [11]. Παρά την ευελιξία και την ερμηνευσιμότητά τους, τα μοντέλα αυτά έχουν περιορισμούς ως προς την απόδοση στη μοντελοποίηση ακολουθιών, καθώς απαιτούν χειροκίνητο ορισμό κόμβων και ακμών και είναι ευαίσθητα στη διαστατικότητα του λανθάνοντος χώρου και ως προς την κλιμάκωση της παραμέτρου β. Παρ' όλα αυτά, παρέχουν πολύτιμες πληροφορίες και μια εννοιολογική βάση που ενθαρρύνει τη χρήση πιο προηγμένων μοντέλων ακολουθιών.

1.2 Πρόταση

1.2.1 Κίνητρα και ερευνητικοί στόχοι

Οι περισσότερες πρόσφατες προσεγγίσεις ανάκτησης βασίζονται σε πολύπλοκα μοντέλα, όπως αυτά που βασίζονται σε Transformers, τα οποία είναι υπολογιστικά δαπανηρά και προσφέρουν περιορισμένη ερμηνευσιμότητα. Ακόμα, οι προσεγγίσεις που χρησιμοποιούν συμβολικές αναπαραστάσεις συνήθως αντιμετωπίζουν τα τραγούδια ως ολόκληρες οντότητες, αγνοώντας την εσωτερική τους δομή και το ότι τα συμφραζόμενα πιθανώς μεταβάλλονται κατά τις μεταβάσεις σε επόμενα τμήματα.

Η παρούσα διπλωματική εργασία διερευνά προσεγγίσεις περισσότερο ερμηνεύσιμες και απλές αξιοποιώντας μοντέλα που χρησιμοποιούν αναδρομή και εκπαιδεύοντάς τα πάνω στο σύνολο δεδομένων Chordonomicon, μια μεγάλης κλίμακας συλλογή διαδοχών συμβολικών συγχορδιών. Το επίκεντρο είναι η αξιολόγηση του κατά πόσον οι πιο ελαφριές αρχιτεκτονικές μπορούν να αποδώσουν ανταγωνιστικά, παρέχοντας ταυτόχρονα ερμηνευσιμότητα μέσω μοντέλων λανθάνουσας απόστασης και ανάλυσης του συνόλου δεδομένων.

Μια βασική καινοτομία είναι η αξιοποίηση της δομής των συγχορδιών, δηλαδή η αποσύνθεσή τους σε ρίζα, ποιότητα μαζί με επεκτάσεις και μπάσο επιτρέπει στα μοντέλα να καταγράφουν τις αρμονικές σχέσεις με μεγαλύτερη λεπτομέρεια. Οι προβλέψεις γίνονται σε επίπεδο τμημάτων και όχι ολόκληρων τραγουδιών, αντανακλώντας τον τρόπο με τον οποίο τα μουσικά συμφραζόμενα αλλάζουν μεταξύ των τμημάτων αυτών. Αυτή η συνδυαστική προσέγγιση καλύπτει ένα κενό που εντοπίστηκε στη βιβλιογραφία.

Στο αρχικό στάδιο της εργασίας αυτής εφαρμόζονται στατιστικές μέθοδοι βασισμένες σε ακολουθίες μήκους n, προσφέροντας ερμηνεύσιμες στατιστικές οπτικές για τη μουσική δομή σε συνδυασμό με μοντέλα λανθάνοντος χώρου. Τα μοντέλα αυτά δημιουργούν έναν λανθάνοντα χώρο όπου οι αποστάσεις αντιστοιχούν σε ομοιότητες των πλειάδων, επιτρέποντας την ανάδειξη μουσικά ουσιαστικών προτύπων για προβλήματα όπως η ταξινόμηση είδους και δεκαετίας. Στη συνέχεια, πραγματοποιούνται πειράματα με ακολουθιακά μοντέλα, βασισμένα στην ανάδραση, για την πρόβλεψη συγχορδιών, αξιοποιώντας αποσυντεθειμένες αναπαραστάσεις συγχορδιών και πληροφορία συμφραζομένων σε επίπεδο τμημάτων.

Με βάση αυτά τα κίνητρα, οι κύριοι στόχοι αυτής της διπλωματικής είναι οι εξής:

- Εκτέλεση στατιστικής ανάλυσης του συνόλου συμβολικών δεδομένων μεγάλης κλίμακας για τον εντοπισμό στοιχείων μιας μουσικά σημαντικής δομής.
- Εφαρμογή ερμηνεύσιμων μοντέλων, όπως μοντέλα λανθάνουσας απόστασης, για την αποκάλυψη αρμονικών και στυλιστικών σχέσεων με βάση στατιστικά στοιχεία ακολουθιών μήκους n.
- Αξιολόγηση ελαφρών αρχιτεκτονικών, συμπεριλαμβανομένων μοντέλων αναδρομής και κατάστασης-χώρου, σε προβλήματα ανάκτησης μουσικών πληροφοριών βασισμένα σε συμβολικές αναπαραστάσεις.
- Εξερεύνηση αποσυντεθειμένων αναπαραστάσεων συγχορδιών (ρίζα, ποιότητα μαζί με επεκτάσεις, μπάσο) και τμηματοποίησης σε επίπεδο τμημάτων για την αξιολόγηση της επίδρασής τους σε προβλήματα πρόβλεψης.
- Σύγκριση των αποτελεσμάτων μεταξύ των επιλογών μοντελοποίησης, ενσωματώνοντας τα ευρήματα από στατιστικές, αναδρομικές και κατάστασης-χώρου προσεγγίσεις.

1.2.2 Επισκόπηση του προτεινόμενου πλαισίου

Η μελέτη αυτή χρησιμοποιεί δύο συμπληρωματικές πειραματικές διαδικασίες έχοντας ως βάση το Chordonomicon.

Στατιστική μοντελοποίηση και μοντελοποίηση λανθάνοντος χώρου για ταξινόμηση

Η πρώτη διαδικασία χρησιμοποιεί ακολουθίες μήκους n, μέγιστου μήκους έξι, για να καταγράψει τοπικές αρμονικές εξαρτήσεις και μοντέλα λανθάνουσας απόστασης για να ενσωματώσει αυτές τις ακολουθίες συγχορδιών σε έναν συνεχή χώρο, όπου οι αποστάσεις αντανακλούν την ομοιότητα τους. Για τη διαχείριση την συνδυαστικής πολυπλοκότητας, όσο αυξάνεται το μήκος των διαδοχών, χρησιμοποιείται μια ατονική αναπαράστασης κάθε συγχορδίας (κρατείται μόνο η ρίζα, τάξη τόνου 0-11). Ακόμα, γίνεται χρήση ελαφρών αρχιτεκτονικών αναδρομικών νευρωνικών δικτύων ως μέτρο σύγκρισης για τα προβλήματα της ταξινόμησης.

Ακολουθιακή μοντελοποίηση για πρόβλεψη συγχορδιών

Η δεύτερη διαδικασία διερευνά τις χρονικές εξαρτήσεις κάνοντας χρήση ακολουθιακών αρχιτεκτονικών για την πρόβλεψη συγχορδιών. Οι συγχορδίες αναπαρίστανται είτε ως μεμονωμένα σύμβολα είτε αποσυντίθενται σε ρίζα, ποιότητα μαζί με επεκτάσεις και μπάσο. Οι προβλέψεις γίνονται σε επίπεδο τμημάτων για να καταγράψουν τα συμφραζόμενα κάθε τμήματος, αντιμετωπίζοντας τους περιορισμούς της μοντελοποίησης ολόκληρου του τραγουδιού.

Γενική Προοπτική

Οι δύο μεθοδολογικές προσεγγίσεις λειτουργούν συμπληρωματικά. Η στατιστική και λανθάνουσα αναπαράσταση εξετάζει τη δομή του συνόλου δεδομένων και τις σχέσεις που προκύπτουν μέσα από αυτό, ενώ η ακολουθιακή μοντελοποίηση εστιάζει στην πρόβλεψη συγχορδιών και στον τρόπο με τον οποίο η διάσπαση τους καθώς και η διάσπαση των τραγουδιών σε επιμέρους στοιχεία επηρεάζει τη μάθηση αρμονικών ακολουθιών.

Ο συνδυασμός των δύο προσεγγίσεων επιδιώκει να ισορροπήσει την ερμηνευσιμότητα της συμβολικής ανάλυσης με την προγνωστική ισχύ των μοντέλων, δείχνοντας ότι αποδοτικές και κατανοητές μέθοδοι μπορούν να αποκαλύψουν ουσιαστική πληροφορία σχετικά με τη δομή της τονικής μουσικής και να στηρίξουν τις αντίστοιχες αναλύσεις και τα πειράματα.

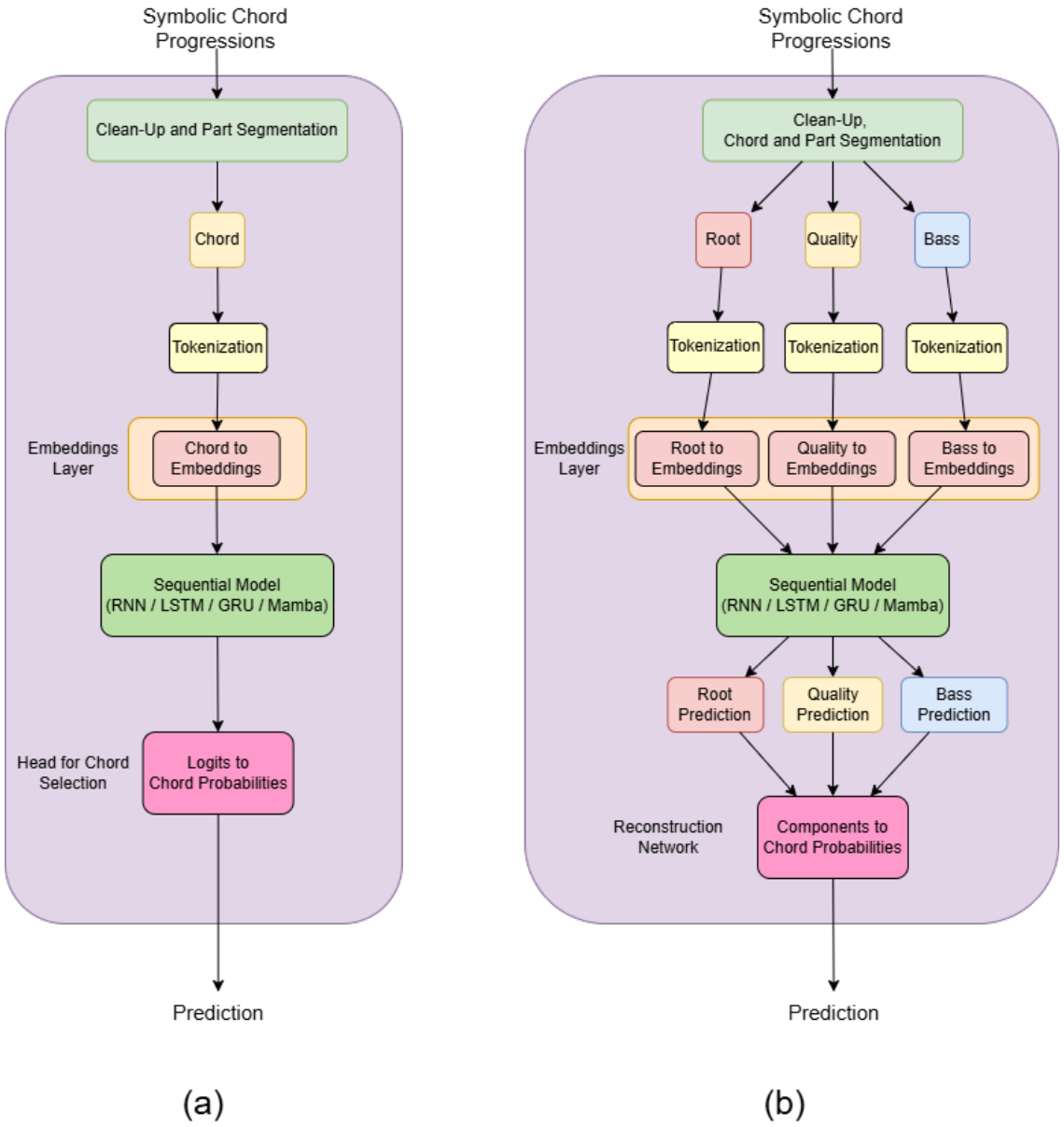


Figure 1.2.1: Οι δύο ροές επεξεργασίας για τα μοντέλα πρόβλεψης συγχορδιών, (a) η προσέγγιση ενός συμβόλου ανά συγχορδία και (b) η προσέγγιση ενός συμβόλου ανά συστατικό της συγχορδίας (3-όροι).

1.3 Σύνολο Δεδομένων και Ανάλυση

1.3.1 Επισκόπηση Συνόλου Δεδομένων

Οι συγχορδίες είναι μείζονος σημασίας για την ανάλυση της μουσικής, και οι πρόσφατες εξελίξεις στην Ανάκτηση Μουσικής Πληροφορίας (ΑΜΠ) έχουν αυξήσει την ανάγκη για μεγάλα, καλά δομημένα σύνολα δεδομένων. Ωστόσο, η δημιουργία τέτοιων πόρων είναι ιδιαίτερα δύσκολη λόγω των περιορισμών πνευματικών δικαιωμάτων και της εγγενούς αμφισημίας της μουσικής αναπαράστασης. Αυτά οδηγούν σε ασυνέπειες στις επισημειώσεις κατά την συλλογή της καθώς τα δεδομένα προέρχονται κυρίως από χρήστες χωρίς επίσημη επιμέλεια.

Το σύνολο δεδομένων Chordonomicon αντιμετωπίζει πολλά από αυτά τα ζητήματα προσφέροντας τη μεγαλύτερη συλλογή συμβολικών διαδοχών συγχορδιών μέχρι σήμερα, με πάνω από 666.000 επισημειωμένα τραγούδια. Τα δεδομένα προέρχονται από το Ultimate Guitar, όπου από τα διπλότυπα τραγουδιών κρατείται το πιο αξιόπιστο μέσω σταθμισμένων αξιολογήσεων. Μια διαδικασία κανονικοποίησης φιλτράρει τις προβληματικές καταχωρήσεις, αφαιρεί τις περιττές επαναλήψεις συγχορδιών ενώ τις μετατρέπει όλες σε τυποποιημένη μορφή μέσω της σύνταξης Harte [12]. Τα δεδομένα του τελικού συνόλου είναι σε συμφωνία επίσης με την Οντολογία Λειτουργικής Αρμονίας [13] και ελέγχονται από ειδικούς που διασφαλίζουν τη μουσική συνέπεια.

Το σύνολο δεδομένων περιλαμβάνει επίσης ετικέτες δομικών τμημάτων που χωρίζουν κάθε τραγούδι, και είναι εμπλουτισμένο με μεταδεδομένα που ανακτώνται μέσω του Spotify API, όπως τη δεκαετία κυκλοφορίας, το είδος μουσικής του καλλιτέχνη και τα σχετικά αναγνωριστικά, όταν αυτά τα στοιχεία είναι διαθέσιμα. Παρόλο που εξακολουθεί να υπάρχει κάποιος θόρυβος μαζί με μη ισορροπημένες καταχωρήσεις μεταξύ των κατηγοριών και ελλείψεις στα μεταδεδομένα, η κλίμακα, η δομή και οι πλούσιες πληροφορίες του συνόλου δεδομένων το καθιστούν έναν πολύτιμο πόρο για την έρευνα και ανάλυση στο πεδίο της μουσικής.

1.3.2 Εξαγωγή Χαρακτηριστικών

Το Chordonomicon συνοδεύεται από μια αρχική Εξερευνητική Ανάλυση Δεδομένων (ΕΑΔ), η οποία διεξήχθη από τους δημιουργούς [14], για να παρέχει μια γενική εικόνα της κατανομής των συγχορδιών και των δομικών ιδιοτήτων. Με βάση αυτό το υπόβαθρο, διεξήγαμε τη δική μας εξερευνητική ανάλυση δεδομένων για να εξαγάγουμε ορισμένα βασικά χαρακτηριστικά από το σύνολο δεδομένων και να παρέχουμε περισσότερες πληροφορίες σχετικά με τη δομή του και τις σχέσεις μεταξύ των δεδομένων. Για να κατανοήσουμε καλύτερα το σύνολο δεδομένων, μετατρέψαμε τις ακολουθίες συγχορδιών από το Chordonomicon σε διάφορες μορφές χαρακτηριστικών, καθεμία από τις οποίες έχει σχεδιαστεί για έναν συγκεκριμένο τύπο ανάλυσης.

Δομικά Χαρακτηριστικά Τραγουδιών

Σε επίπεδο τραγουδιού, χρησιμοποιήθηκαν μεταδεδομένα σχετικά με το είδος, τη δεκαετία κυκλοφορίας και τα αναγνωριστικά των καλλιτεχνών, για να καταγραφεί η κατανομή των τραγουδιών και των καλλιτεχνών σε όλες τις κατηγορίες. Σε επίπεδο μερών, εξάγαμε δομικά μέτρα, όπως τον αριθμό των τμημάτων ανά τραγούδι και το μέγεθος των τμημάτων συνολικά και σε σχέση με το είδος και τη δεκαετία. Αυτά τα χαρακτηριστικά δεν εξαρτώνται από την ταυτότητα των συγχορδιών, αλλά υποστηρίζουν τις δομικές αναλύσεις.

Χαρακτηριστικά Αποσύνθεσης

Η σύνταξη του συνόλου δεδομένων επιτρέπει την εύκολη ανάλυση κάθε συγχορδίας στα βασικά της συστατικά, τη ρίζα, την ποιότητα, τις επεκτάσεις και τη νότα του μπάσου. Αυτή η αναπαράσταση επιτρέπει την ανάλυση της αρμονικής πολυπλοκότητας και της στυλιστικής χρήσης, εστιάζοντας ξεχωριστά σε κάθε μέρος μιας συγχορδίας. Μια τέτοια ανάλυση περιλαμβάνει τον τρόπο με τον οποίο χρησιμοποιούνται οι ποιότητες, οι επεκτάσεις και το μπάσο σε διαφορετικά είδη και δεκαετίες, παρέχοντας πληροφορίες για τις στυλιστικές και αρμονικές αποφάσεις στη μουσική σύνθεση. Επιπλέον, οι πληροφορίες για τη ρίζα μπορούν να αξιοποιηθούν για τη μελέτη της αρμονικής ροής μέσω των κινήσεων της ρίζας σε ημιτόνια. Κάθε ρίζα αντιπροσωπεύεται από έναν αριθμό με βάση τη θέση της στη χρωματική κλίμακα (0-11). Μετατρέποντας κάθε ρίζα στην ατονική της αναπαράσταση, οι πληροφορίες σχετικά με την απόλυτη τονικότητα αγνοούνται, ώστε να εστιάσουμε στο σχετικό αρμονικό πλαίσιο και στις σχέσεις μεταξύ των ριζών.

Χαρακτηριστικά Πλήρων Συγχορδιών

Για την αρμονική ανάλυση λεξιλογίου, κάθε συγχορδία διατηρείται στην πλήρη μορφή της, με την εξαίρεση ότι οι ρητές πληροφορίες για το μπάσο (αναστροφή) παραλείπονται, ώστε να δοθεί έμφαση στην ποικιλία των τύπων συγχορδιών και όχι στις επιλογές σχετικά με τις φωνές. Συγκεντρώνονται στατιστικά στοιχεία σχετικά με τον αριθμό των μοναδικών συγχορδιών ανά δεκαετία και είδος, μαζί με τα πιο συνηθισμένες δυάδες και τριάδες σε όλα τα τραγούδια. Εδώ λαμβάνεται επίσης υπόψη η δομή των μερών, λαμβάνοντας υπόψη και τις πλειάδες που χωρίζονται μεταξύ των ορίων των μερών. Για στατιστικά στοιχεία μεταξύ των τμημάτων, καταγράφουμε ρητά τις πλειάδες των οποίων η τελευταία συγχορδία είναι επίσης η πρώτη συγχορδία του επόμενου τμήματος, επιτρέποντας τη μελέτη των μοτίβων μεταξύ των τμημάτων του τραγουδιού.

Η προσέγγιση του μοντέλου λανθάνουσας απόστασης ενσωματώνει ατονικές αναπαραστάσεις ρίζας για κάθε συγχορδία, προκειμένου να αποφευχθεί η συνδυαστική πολυπλοκότητα των πλειάδων καθώς αυξάνεται το n. Η μοντελοποίηση επικεντρώθηκε σε πλειάδες μήκους 4, 5 και 6, οι οποίες εξήχθησαν από το σύνολο δεδομένων. Για να μειωθεί η μεροληψία κατηγορίας, οι πλειάδες επιλέχθηκαν από τραγούδια μετά την εξισορρόπηση του συνόλου δεδομένων σε σχέση με το είδος και τη δεκαετία. Κάθε πλειάδα αντιμετωπίστηκε ως κόμβος γραφήματος, με άκρες που ορίστηκαν μέσω των πλησιέστερων γειτόνων στο χώρο χαρακτηριστικών. Τα γραφήματα που προέκυψαν αποτέλεσαν τις εισόδους για τα μοντέλα λανθάνουσας απόστασης, παράγοντας λανθάνουσες θέσεις σε ένα δισδιάστατο χώρο. Αυτή η διαδικασία περιγράφεται αναλυτικά στην ενότητα 1.3.4.

1.3.3 Περιγραφική Ανάλυση

Με βάση την εξαγωγή χαρακτηριστικών που περιγράφηκε παραπάνω, σε αυτή την ενότητα παρουσιάζουμε τα αποτελέσματα της Εξερευνητικής Ανάλυσης Δεδομένων (ΕΑΔ) που πραγματοποιήσαμε στο σύνολο δεδομένων Chordonomicon. Η ανάλυση επικεντρώνεται τόσο στα στατικά όσο και στα δυναμικά χαρακτηριστικά των συνόλων δεδομένων, παρέχοντας πληροφορίες σχετικά με τις δομικές ιδιότητες και τα ακολουθιακά χαρακτηριστικά.

Οι δομικές ιδιότητες παρέχουν πληροφορίες σχετικά με την οργάνωση των τραγουδιών και των μερών, καθώς και το αρμονικό λεξιλόγιο, συμπεριλαμβανομένων των ιδιοτήτων των συγχορδιών, των επεκτάσεων και της χρήσης μοναδικών συγχορδιών. Επιπλέον, τα ακολουθιακά χαρακτηριστικά καταγράφουν την αρμονική κίνηση και τις σχέσεις μεταξύ γειτονικών στοιχείων, συμπεριλαμβανομένων των διαστημάτων κινήσεων των ριζών και των κοινών μοτίβων πλειάδων. Αυτές οι αναλύσεις επικεντρώνονται στις τάσεις, τις στυλιστικές διαφορές και τις κατανομές μεταξύ των ειδών και των δεκαετιών, λαμβάνοντας επίσης υπόψη τον τρόπο με τον οποίο αυτά τα μοτίβα αλλάζουν γύρω από τα όρια των τμημάτων. Αποτελούν τη βάση για την λανθάνουσα μοντελοποίηση και τις διαφορετικές προσεγγίσεις στα προβλήματα ανάκτησης μουσικών πληροφοριών που παρουσιάζονται στις ενότητες 1.3.4 και 1.4.

Δομικά Χαρακτηριστικά

Ξεκινάμε με μια επισκόπηση των ποσοτικών χαρακτηριστικών του συνόλου δεδομένων και της κάλυψης των μεταδεδομένων, που παρουσιάζονται στο 1.1, προκειμένου να κατανοήσουμε καλύτερα το εύρος και τη σύνθεσή του. Το σύνολο δεδομένων περιέχει πάνω από 679.000 μουσικά κομμάτια που συνολικά έχουν σχεδόν 52 εκατομμύρια συγχορδίες. Το αρμονικό του λεξιλόγιο αποτελείται από 749 συγχορδίες και 3.577 ανεστραμμένες μορφές αυτών. Περίπου 397.000 από τα συμπεριλαμβανόμενα κομμάτια συνοδεύονται επίσης από πληροφορίες σχετικά με τη δομική τους κατάτμηση (πώς χωρίζονται σε τμήματα), που ανέρχονται σε περισσότερα από 2,6 εκατομμύρια μεμονωμένα τμήματα. Επιπλέον, λαμβάνοντας υπόψη τις πρόσθετες επισημειώσεις των μεταδεδομένων, η σύνδεση με τα δεδομένα του Spotify είναι σε μεγάλο βαθμό πλήρης, με διαθέσιμα στοιχεία για το είδος, την ημερομηνία κυκλοφορίας, το τραγούδι και τον καλλιτέχνη για την πλειονότητα των κομματιών.

Table 1.1: Συνοπτικά στατιστικά στοιχεία του συνόλου δεδομένων [14].

Περιγραφή	**Πλήθος**
Αριθμός των Τραγουδιών	679,807
Αριθμός των Συγχορδιών	51,994,634
Αριθμός των Μοναδικών Συγχορδιών	749
Αριθμός των Μοναδικών Ανεστραμένων Συγχορδιών	3,577
Αριθμός των Τραγουδιών με Τμήματα	397,580
Αριθμός των Parts	2,670,457
Αριθμός των Τραγουδιών με Release Date	422,181
Αριθμός των Τραγουδιών με Spotify Genres	429,753
Αριθμός των Τραγουδιών με Main Genre	352,111
Αριθμός των Τραγουδιών με Spotify Track ID	440,284
Αριθμός των Τραγουδιών με Spotify Artist ID	510,986

Μετά από αυτή την επισκόπηση, εξετάζουμε την κατανομή των τραγουδιών και των καλλιτεχνών στις δώδεκα κύριες κατηγορίες μουσικών ειδών. Τα ιστογράμματα που απεικονίζουν τον αριθμό των μοναδικών τραγουδιών, των καλλιτεχνών και την αναλογία μεταξύ τους παρουσιάζονται στο Σχήμα 1.3.1. Η κατανομή των τραγουδιών ανά μουσικό είδος είναι εμφανώς μη ισορροπημένη, με μουσικά είδη όπως η Ποπ, η Ροκ, η Εναλλακτική, η Κάντρι και η Ποπ-Ροκ να αντιπροσωπεύουν συνολικά την πλειονότητα των τραγουδιών που έχουν αντιστοιχιστεί σε ένα κύριο μουσικό είδος. Ομοίως, η κατανομή των καλλιτεχνών ανά είδος είναι ασύμμετρη, με τα ίδια είδη να αντιπροσωπεύουν περισσότερα από τα δύο τρίτα όλων των μοναδικών καλλιτεχνών. Είναι σημαντικό να σημειωθεί ότι αυτές οι δύο κατανομές δεν είναι άμεσα ανάλογες. Η αναλογία τραγουδιών ανά καλλιτέχνη αντικατοπτρίζει τη σχετική εκπροσώπηση των καλλιτεχνών σε κάθε είδος στο σύνολο δεδομένων. Για παράδειγμα, οι καλλιτέχνες της Ποπ-Ροκ συνεισφέρουν κατά μέσο όρο σχεδόν είκοσι τραγούδια, ενώ οι καλλιτέχνες της Κάντρι και της Ροκ συνεισφέρουν κατά μέσο όρο περισσότερα από δέκα τραγούδια ανά καλλιτέχνη. Τα περισσότερα άλλα είδη έχουν κατά μέσο όρο περίπου έξι τραγούδια ανά καλλιτέχνη, με τη Ραπ και την Ηλεκτρονική να παρουσιάζουν ακόμη χαμηλότερες τιμές.

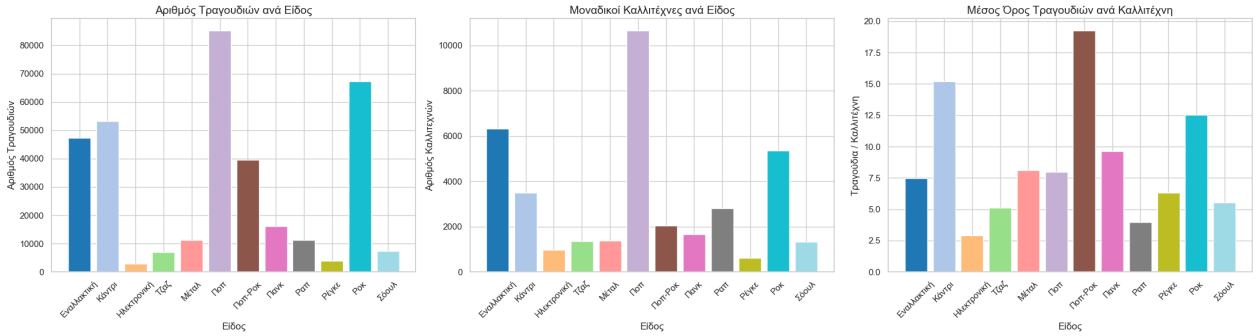
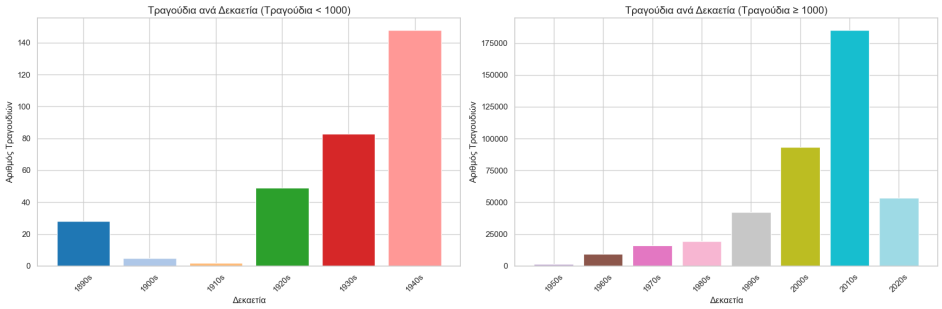


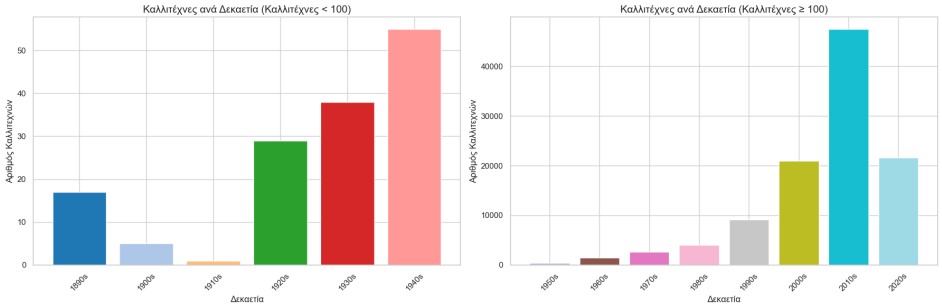
Figure 1.3.1: Κατανομή τραγουδιών και καλλιτεχνών ανά είδος και μέσος αριθμός τραγουδιών ανά καλλιτέχνη.

Για να συμπληρωθούν τα παραπάνω στατιστικά στοιχεία, το Σχήμα 1.3.2 δείχνει την κατανομή των κομματιών και των καλλιτεχνών ανά δεκαετία. Κάθε κατανομή χωρίζεται σε δύο μέρη λόγω της δυσανάλογης κατανομής των τραγουδιών και των καλλιτεχνών στις προηγούμενες δεκαετίες σε σύγκριση με τις πιο πρόσφατες. Η κατανομή των τραγουδιών ανά δεκαετία έχει το πρώτο γράφημα να δείχνει δεκαετίες με λιγότερα από 1.000 τραγούδια, ενώ το πρώτο γράφημα για την κατανομή των καλλιτεχνών ανά δεκαετία δείχνει εκείνες τις δεκαετίες με λιγότερους από 100 καλλιτέχνες να τις εκπροσωπούν. Τα δεύτερα γραφήματα δείχνουν τις δεκαετίες που ξεπερνούν αυτά τα όρια. Αυτός ο διαχωρισμός επιτρέπει τη σαφή οπτικοποίηση τόσο των αραιών όσο και των πυκνών περιόδων. Τα περισσότερα τραγούδια και καλλιτέχνες στο σύνολο δεδομένων προέρχονται από τη δεκαετία του 2010, ακολουθούμενα από τη δεκαετία του 2000 και του 2020, που συνολικά περιλαμβάνουν την πλειονότητα των εγγραφών. Οι λιγότερο αντιπροσωπευόμενες δεκαετίες είναι αυτές πριν από τη δεκαετία του 1950, με τις δεκαετίες του 1910 και του 1900 να είναι οι πιο αραιές. Σε αντίθεση με τις κατανομές σε επίπεδο είδους, οι κατανομές τραγουδιών ανά δεκαετία και καλλιτεχνών ανά δεκαετία είναι σε μεγάλο βαθμό αναλογικές, με τη δεκαετία του 2020 να αποτελεί εξαίρεση λόγω του μικρότερου αριθμού κομματιών, παρά τον μέτριο αριθμό

καλλιτεχνών που συνέβαλαν.



(a) Song Distribution by Decade.



(b) Artist Distribution by Decade.

Figure 1.3.2: Κατανομή τραγουδιών και καλλιτεχνών ανά δεκαετία.

Συνεχίζοντας την ανάλυση των σχέσεων μεταξύ των τραγουδιών που επιλέχθηκαν από διάφορα είδη και δεκαετίες, στο Σχήμα 1.3.3, τα τραγούδια που έχουν τόσο ένα κύριο είδος όσο και ημερομηνία κυκλοφορίας χρησιμοποιούνται για να παρουσιάσουν την κατανομή κάθε είδους ανά δεκαετία. Σύμφωνα με την κατανομή των τραγουδιών ανά δεκαετία, η πλειονότητα των τραγουδιών σε κάθε είδος ανήκει στη δεκαετία του 2010, με τη δεκαετία του 2000 να ακολουθεί σε μικρή απόσταση σε είδη όπως η Κάντρι, η Τζαζ, το Μέταλ, η Ποπ-Ροκ, η Πάνκ, η Ρέγκε και η Ροκ, ενώ έχει μικρότερη εκπροσώπηση στα υπόλοιπα. Επιπλέον, τα περισσότερα είδη έχουν μικρή εκπροσώπηση στις δεκαετίες πριν από τη δεκαετία του 1990, με αξιοσημείωτες εξαιρέσεις τα είδη Κάντρι, Τζαζ, Ποπ-Ροκ, Ρέγκε και Ροκ, όπου η εκπροσώπηση που έχει συσσωρευτεί σε αυτές τις δεκαετίες είναι συγκρίσιμη με τα υπόλοιπα.

Προχωρώντας από τα μεταδεδομένα για ολόκληρες καταχωρήσεις τραγουδιών, το Σχήμα 1.3.4 παρουσιάζει κατανομές με βάση δομικά τμήματα, που προέρχονται από καταχωρήσεις του συνόλου δεδομένων που περιλαμβάνουν πληροφορίες για τα τμήματα. Το πρώτο διάγραμμα δείχνει την κατανομή του αριθμού των τμημάτων ανά τραγούδι, που μοιάζει με μια κανονική κατανομή με μέσο όρο το 7, με τον αριθμό των τραγουδιών να μειώνεται εκθετικά καθώς αυξάνονται τα τμήματα, και τελικά να γίνεται ασήμαντος πέρα από τα 15 τμήματα.

Στο δεύτερο και τρίτο γράφημα παρουσιάζεται ο μέσος αριθμός τμημάτων σε ένα τραγούδι σε σχέση με τα είδη και τις δεκαετίες. Και στα δύο γραφήματα, οι περισσότεροι μέσοι όροι κυμαίνονται μεταξύ 6 και 8 τμημάτων σε ένα τραγούδι, με τη Μέταλ να έχει λίγο υψηλότερο μέσο όρο και τη Ραπ να έχει τον χαμηλότερο μέσο όρο μεταξύ των ειδών, ενώ ο μέσος όρος της δεκαετίας του 1890 ξεπερνά τα 8 τμήματα, ακολουθούμενος από τη δεκαετία του 1980 με μέσο όρο περίπου 7 τμήματα ανά τραγούδι.

Εξερευνώντας ένα επίπεδο βαθύτερα, το Σχήμα 1.3.5 απεικονίζει την κατανομή των μεγεθών των τμημάτων μετρημένα σε αριθμό συγχορδιών. Τα μεγέθη τμημάτων μεγαλύτερα από 50 εξαιρέθηκαν λόγω της αμελητέας συχνότητάς τους. Στο πρώτο διάγραμμα, η συνολική κατανομή κορυφώνεται γύρω στις 8 συγχορδίες ανά τμήμα, ενώ τοπικές κορυφές παρατηρούνται στα μήκη 4, 6, 12 και 16 και μερικές μικρότερες στα 20 και 24. Είναι σημαντικό να σημειωθεί ότι πολλές από αυτές τις τοπικές κορυφές εμφανίζονται σε πολλαπλάσια του 4.

Τα μεγέθη των τμημάτων σε διάφορα είδη και δεκαετίες μπορούν να παρατηρηθούν στο δεύτερο και τρίτο γράφημα. Το μέσο μέγεθος των τμημάτων κυμαίνεται μεταξύ 12 και 14 και για τα δύο γραφήματα. Στο γράφημα

Figure 1.3.3: Κατανομή των ειδών ανά δεκαετία.

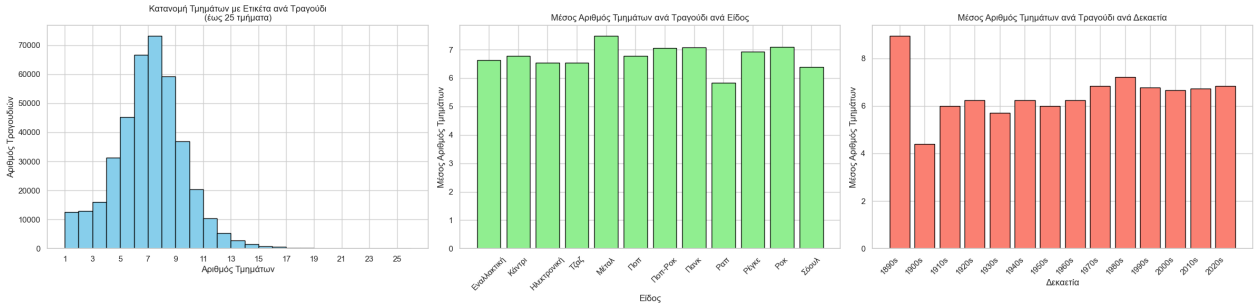


Figure 1.3.4: Αριθμός τμημάτων ανά τραγούδι συνολικά, ανά είδος και ανά δεκαετία.

των ειδών, αυτό που είναι αξιοσημείωτο είναι ότι το μέσο μέγεθος των τμημάτων της Ραπ είναι υψηλότερο από τα υπόλοιπα, ακόμη και αν η διαφορά είναι αμελητέα, ενώ προηγουμένως η Ραπ παρουσίαζε το μικρότερο μέσο όρο στον αριθμό των τμημάτων ανά τραγούδι. Λαμβάνοντας υπόψη την κατανομή του μεγέθους των τμημάτων ανά δεκαετία, η δεκαετία του 1900 αποτελεί εξαίρεση, με μέσο όρο υψηλότερο από 16 συγχορδίες ανά τμήμα, ενώ η δεκαετία του 1950 έχει τη μικρότερη τιμή, περίπου 11 συγχορδίες ανά τμήμα.

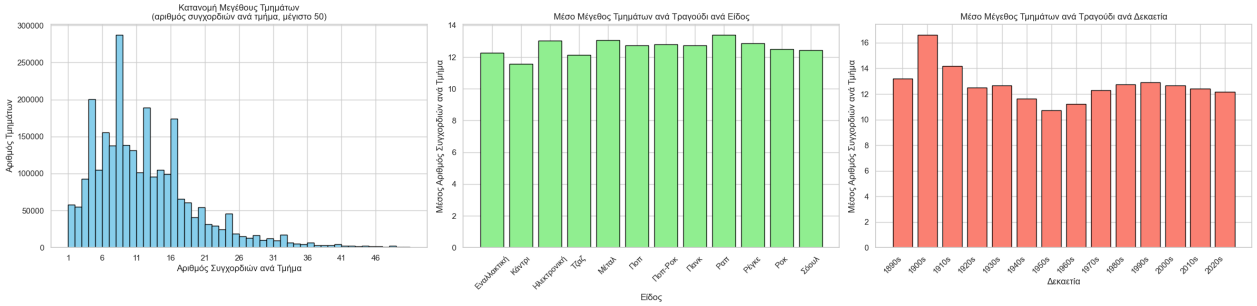
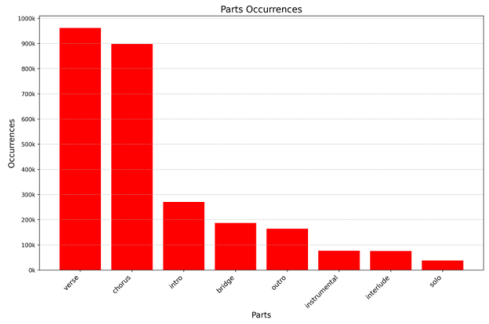
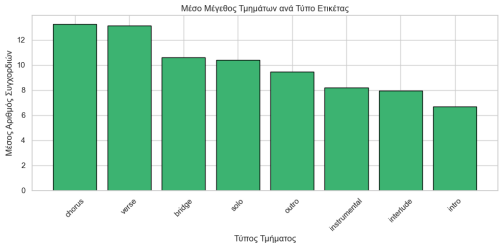


Figure 1.3.5: Μέγεθος τμημάτων συνολικά, ανά είδος και ανά δεκαετία.

Σύμφωνα με τα προηγούμενα διαγράμματα, το Σχήμα 1.3.6 παρουσιάζει επιπλέον κατανομές σχετικά με τους τύπους τμημάτων και το μέσο μήκος ανά τύπο. Στο πρώτο γράφημα, όπου παρουσιάζεται η κατανομή των τύπων των τμημάτων, είναι προφανές ότι οι Στροφές και τα Ρεφρέν υπερτερούν αριθμητικά των υπόλοιπων τύπων τμημάτων, ενώ τη δεύτερη θέση καταλαμβάνουν οι Εισαγωγές, οι Γέφυρες και οι Καταλήξεις, ακολουθούμενες από τα Οργανικά, τα Ενδιάμεσα μέρη και τα Σόλο, με λιγότερες από 100.000 εμφανίσεις το καθένα. Το δεύτερο διάγραμμα εμφανίζει το μέσο μήκος των τμημάτων ανά τύπο. Οι μέσοι όροι κυμαίνονται μεταξύ 6 και 14 συγχορδιών, με τα Ρεφρέν και τις Στροφές να έχουν μέσο όρο πάνω από 12 και τις Εισαγωγές να είναι τα πιο σύντομα με μέσο όρο περίπου 6 συγχορδίες ανά εμφάνιση. Οι δύο κατανομές φαίνεται να έχουν τους τύπους τμημάτων στην ίδια σειρά, με εξαίρεση τις Εισαγωγές που μετακινήθηκαν στην τελευταία θέση.



(a) Κατανομή μεγεθών τμημάτων (αριθμός συγχορδιών ανά τμήμα) [14].



(b) Μέσο μήκος τμήματος ανά τύπο ετικέτας.

Figure 1.3.6: Σύγκριση στατιστικών στοιχείων για τη δομή των χορδών: (α) κατανομή μεγέθους τμημάτων και (β) μέσο μήκος ανά τύπο τμήματος.

Το επόμενο βήμα σε αυτή την ανάλυση είναι να προχωρήσουμε στο επίπεδο των συγχορδιών. Στο Σχήμα 1.3.7 παρουσιάζεται η κατανομή των μοναδικών συγχορδιών ανά τραγούδι σε όλα τα είδη και τις δεκαετίες. Σε όλα τα είδη είναι αξιοσημείωτο ότι η Τζαζ και η Σόουλ παρουσιάζουν τους υψηλότερους μέσους όρους μοναδικών συγχορδιών και τυπικής απόκλισης ανά τραγούδι, ενώ τα υπόλοιπα χωρίζονται σε δύο ομάδες: αυτά με μέσους όρους άνω των 6 μοναδικών συγχορδιών και τυπική απόκλιση κοντά στα κορυφαία, δηλαδή η Μέταλ, η Ποπ, η Ποπ-Ροκ και η Ροκ, και αυτά με μέσους όρους κάτω των 6 μοναδικών συγχορδιών και μικρή τυπική απόκλιση, δηλαδή η Εναλλακτική, η Κάντρι, η Ηλεκτρονική, η Πάνκ, η Ραπ και η Ρέγκε. Μεταξύ των δεκαετιών, τόσο οι μέσοι όροι όσο και η τυπική απόκλιση είναι κοντά μεταξύ τους, με κορυφή κατά τη δεκαετία του 1920 και συνολική μείωση προς τη δεκαετία του 2020, χωρίς πολλές αξιοσημείωτες αλλαγές. Όσον αφορά τις δεκαετίες πριν από τη δεκαετία του 1920, παρουσιάζουν μικρότερες τιμές, τόσο στους μέσους όρους όσο και στις τυπικές αποκλίσεις, σε σύγκριση με τις υπόλοιπες.

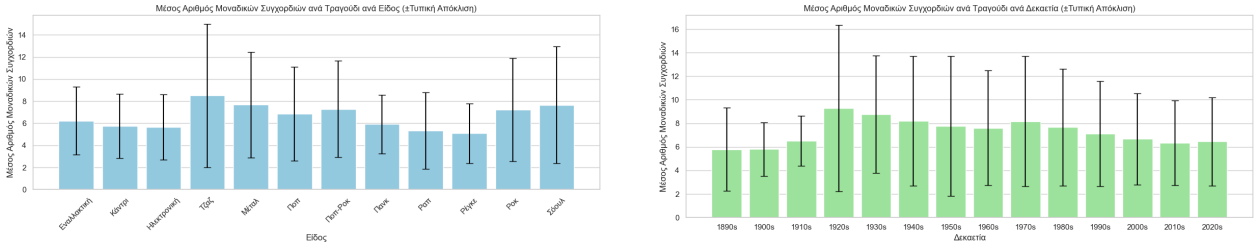


Figure 1.3.7: Μέση χρήση μοναδικών συγχορδιών ανά είδος και δεκαετία.

Περισσότερες πληροφορίες για τις συγχορδίες παρουσιάζονται στο Σχήμα 1.3.8, όπου οι πιο συχνές συγχορδίες παρουσιάζονται μαζί με τα ποσοστά εμφάνισής τους. Είναι προφανές ότι αυτές οι συγχορδίες, απλές στη μορφή τους, κυριαρχούν στο σύνολο των δεδομένων, χωρίς επεκτάσεις ή αναστροφές, χρησιμοποιώντας τις πιο κοινές ιδιότητες μείζονα (G, C, D, A, F, E) και ελάσσονα (Amin, Emin, Bmin). Μαζί, αυτές οι 9 κορυφαίες επιλογές αποτελούν περίπου τα δύο τρίτα της συνολικής χρήσης συγχορδιών.

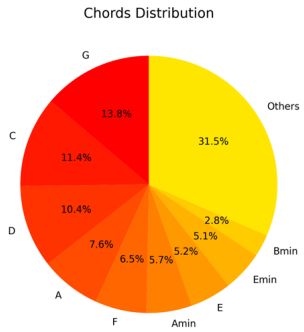
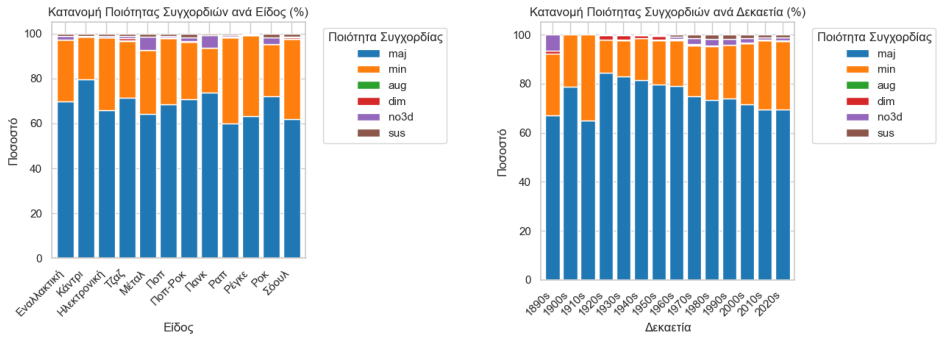


Figure 1.3.8: Κατανομή χρήσης συγχορδιών [14].

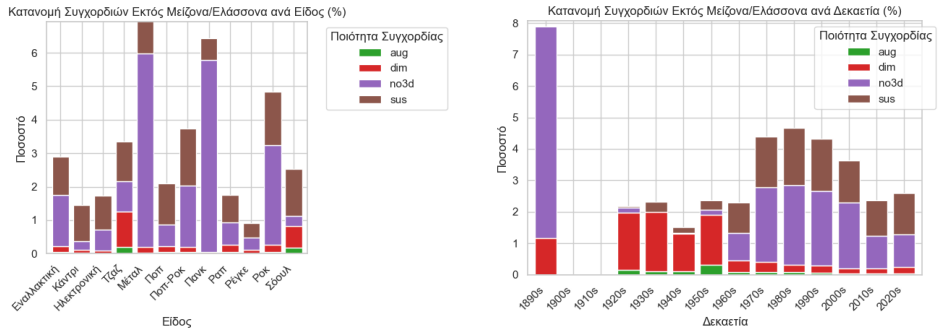
Στη συνέχεια, η ανάλυση επικεντρώνεται στα συστατικά στοιχεία των συγχορδιών, δηλαδή στις ποιότητες, τις επεκτάσεις και τις αναστροφές. Ξεκινώντας από τις ποιότητες, η κατανομή τους παρουσιάζεται στο Σχήμα 1.3.9. Το πάνω μέρος του σχήματος δείχνει την κατανομή των ποιοτήτων στα διάφορα είδη και δεκαετίες. Είναι σαφές ότι η πλειοψηφία αποτελείται από μείζονες συγχορδίες, με τις ελάσσονες συγχορδίες να είναι ο δεύτερος πιο σημαντικός τύπος ποιότητας και μαζί να αποτελούν πάνω από το 90% σε όλες τις κατηγορίες. Η αναλογία των μείζονων και ελάσσονων τύπων αλλάζει ελαφρώς μεταξύ των ειδών, ενώ φαίνεται ότι οι ελάσσονες συγχορδίες καλύπτουν περισσότερο χώρο από τη δεκαετία του 1920 έως τη δεκαετία του 2020.

Για να διερευνηθούν οι ακραίες καταστάσεις, οι κατανομές τους εμφανίζονται στο κάτω μέρος του Σχήματος 1.3.9. Όπως φαίνεται σε αυτά τα διαγράμματα, στα περισσότερα είδη αυτές οι ανώμαλες ποιότητες δεν υπερβαίνουν το 5%, με εξαιρέσεις τα είδη Μέταλ, Πάνκ και Ροκ, όπου κυριαρχούν οι Δυναμικές συγχορδίες. Οι Δυναμικές συγχορδίες φαίνεται να κυριαρχούν στο μισό των ειδών, ενώ στο άλλο μισό είτε υπάρχουν περισσότερες Αναστολές είτε είναι ίσες. Οι Αυξημένες και Μειωμένες ποιότητες εμφανίζονται σε ασήμαντο βαθμό εκτός από τα είδη Τζαζ και Σόουλ, όπου οι Μειωμένες συγχορδίες έχουν συγκρίσιμες αναλογίες με τις Δυναμικές και τις Ανασταλμένες συγχορδίες. Όσον αφορά τις κατανομές κατά τις δεκαετίες, οι Μειωμένες συγχορδίες φαίνεται να είναι πιο δημοφιλείς πριν από τη δεκαετία του 1960, με τις Δυναμικές και τις Ανασταλμένες να κυριαρχούν μετά από αυτή την περίοδο.

Στη συνέχεια, εξετάζουμε τις εμφανίσεις αναστροφών στο σύνολο δεδομένων. Όπως παρουσιάζεται στο πάνω μέρος του Σχήματος 1.3.10, η κατανομή των επιλογών αναστροφών παραμένει σταθερή σε όλα τα είδη και τις δεκαετίες, με τα $B, F\#$ και G να επιλέγονται ελαφρώς περισσότερο ως νότες μπάσου. Ορισμένες ασυνέπειες εμφανίζονται στις κατανομές κατά τις δεκαετίες πριν από τη δεκαετία του 1920, δείχνοντας την πλήρη κυριαρχία συγκεκριμένων νότων στο μπάσο, όπως το E κατά τη δεκαετία του 1910, το B κατά τη δεκαετία του 1890 και ένα κενό κατά τη δεκαετία του 1900.



(a)



(b)

Figure 1.3.9: Κατανομή (α) όλων των ποιοτήτων και (β) των σπάνιων ποιοτήτων, ανά είδος και δεκαετία.

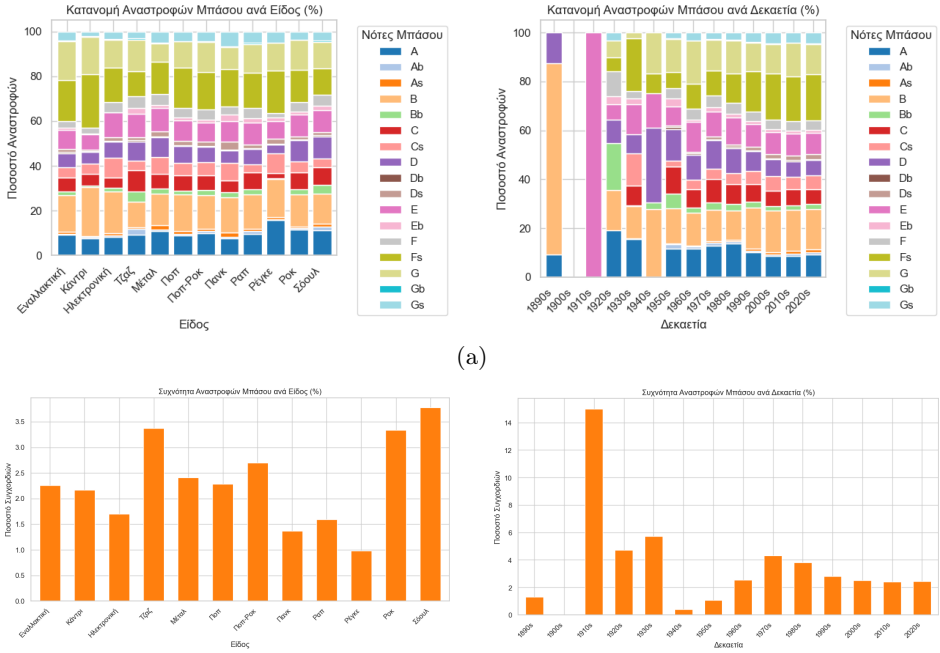
Το κάτω μέρος του Σχήματος 1.3.10 δείχνει τα ποσοστά χρήσης αναστροφών στις διάφορες κατηγορίες. Σε όλα τα είδη, τα ποσοστά δεν διαφέρουν πολύ, με τα υψηλότερα να είναι η Τζαζ, η Ροκ και η Σόουλ με περίπου 3.5%, ενώ τα χαμηλότερα ποσοστά αντιστοιχούν στην Πάνκ και τη Ρέγκε με περίπου 1%. Κατά τη διάρκεια των δεκαετιών, είναι προφανές ότι μια τοπική κορύφωση εμφανίζεται γύρω στη δεκαετία του 1970, μειούμενη ελαφρώς προς τα τελευταία χρόνια, ενώ τα προηγούμενα ποσοστά εμφανίζονται επίσης χαμηλά, εκτός από τη δεκαετία του 1910, όπου εκτοξεύονται στο 14%, και τις δεκαετίες του 1920 και 1930, που έχουν παρόμοιες εμφανίσεις αναστροφών με τη δεκαετία του 1970.

Τέλος, οι κατανομές των επεκτάσεων εμφανίζονται στο Σχήμα 1.3.11. Και στα δύο διαγράμματα μπορεί να παρατηρηθεί ότι η πιο συνηθισμένη επέκταση είναι η προσθήκη της 7ης νότας στη συγχορδία, με σημαντική διαφορά από την προσθήκη της 9ης, η οποία καταλαμβάνει τη δεύτερη θέση για την πλειονότητα των κατηγοριών, ενώ φαίνεται επίσης ότι η 13η είναι πιο δημοφιλής επιλογή από την 11η. Οι πιο αξιοσημείωτες παρατηρήσεις είναι ότι η Τζαζ και η Σόουλ φαίνεται να χρησιμοποιούν τις επεκτάσεις περισσότερο από τα υπόλοιπα είδη, ενώ η Μέταλ και η Πάνκ τις χρησιμοποιούν λιγότερο. Λαμβάνοντας υπόψη τις δεκαετίες, η χρήση των επεκτάσεων φαίνεται να κορυφώνεται γύρω στη δεκαετία του 1920, με χαμηλότερα ποσοστά πριν από αυτή, και να μειώνεται σταδιακά καθώς πλησιάζουμε στην εποχή μας.

Ακολουθιακά Χαρακτηριστικά

Σε αυτή την ενότητα, παρουσιάζουμε τα στατιστικά στοιχεία που προέκυψαν από την ανάλυση που επικεντρώθηκε στις τοπικές εξαρτήσεις στις ακολουθίες συγχορδιών, που σημαίνει ότι χρησιμοποιούνται δυάδες και τριάδες συγχορδιών, ενώ πλειάδες υψηλότερης τάξης, συγκεκριμένα τετράδες, πεντάδες και εξάδες χρησιμοποιούνται στο 1.3.4.

Πρώτον, στο Σχήμα 1.3.12 παρουσιάζουμε την κατανομή των κινήσεων των ριζών των νότων μεταξύ των μεταβάσεων των συγχορδιών. Και στα δύο διαγράμματα, το ένα που περιγράφει τις κινήσεις σε όλα τα είδη και το δεύτερο σε όλες τις δεκαετίες, παρατηρούμε ότι υπάρχει μια προτίμηση στις κινήσεις των 5 ημιτόνων, ακολουθούμενη από τις κινήσεις των 2 ημιτόνων, με την πρώτη να είναι σχεδόν διπλάσια σε ποσοστό από τη δεύτερη. Οι κινήσεις 3 και 4 ημιτόνια φαίνεται να επιλέγονται εξίσου, ενώ οι υπόλοιπες βρίσκονται σε χαμηλότερα επίπεδα, με τις κινήσεις 6 ημιτόνια να είναι οι πιο σπάνιες και σχεδόν αμελητέες. Μια άλλη αξιοσημείωτη



(a)

(b)

Figure 1.3.10: Κατανομή (α) των επιλογών αντιστροφής και (β) των ποσοστών ύπαρξης αντιστροφής, ανά είδος και δεκαετία.

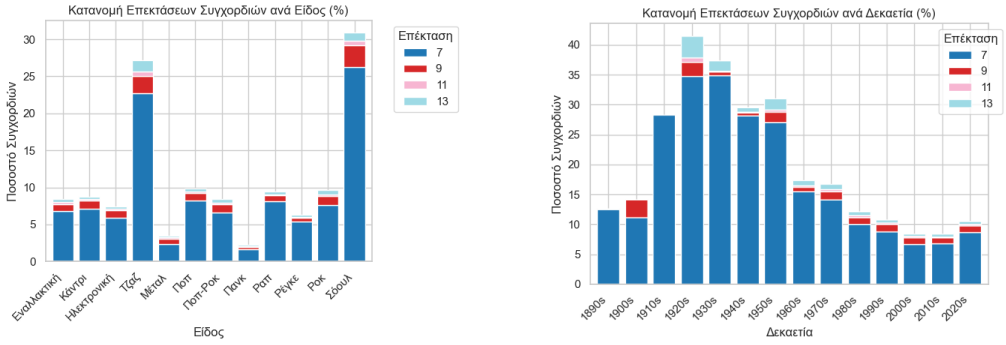


Figure 1.3.11: Κατανομή των επεκτάσεων ανά είδος και δεκαετία.

παρατήρηση είναι ότι η προτίμηση μεταξύ της κίνησης προς τα πάνω ή προς τα κάτω σε ημιτόνια φαίνεται να είναι ελάχιστη σε όλες τις κατηγορίες, με τη μεγαλύτερη διαφορά να είναι περίπου 2% για την κίνηση κατά 2 ημιτόνια.

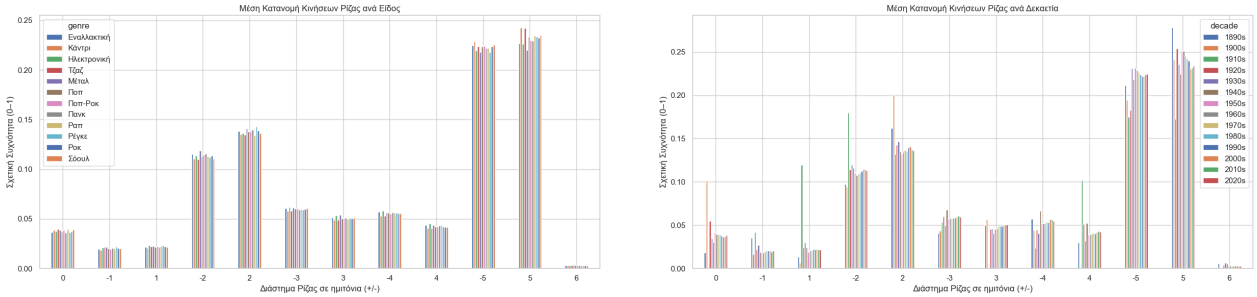


Figure 1.3.12: Κινήσεις ρίζας μετρημένες σε ημιτόνια στα διάφορα είδη και δεκαετίες.

Προχωρώντας με την ακολουθιακή ανάλυση, συλλέξαμε τις πιο συνηθισμένες δυάδες και τριάδες συγχορδιών, εστιάζοντας σε κομμάτια με έγκυρες ετικέτες τμημάτων. Οι 20 πιο συνηθισμένες εμφανίζονται στο Σχήμα 1.3.13, όπου για κάθε πλειάδα παρουσιάζονται δύο στήλες, η πρώτη με τις πιο συνηθισμένες πλειάδες συνολικά και η δεύτερη με τις πιο συνηθισμένες στις μεταβάσεις τμημάτων. Οι πλειάδες μεταξύ των τμημάτων έχουν την τελευταία συγχορδία τους να ανήκει στο επόμενο τμήμα του τραγουδιού, ως τρόπος μέτρησης του πώς αλλάζει το πλαίσιο μεταξύ των τμημάτων. Είναι εμφανές και στους δύο πίνακες ότι η κατανομή των κορυφαίων συχνοτήτων χάνει τη δομή της σε σύγκριση με τις συχνότητες στα όρια των τμημάτων. Η πιο κοινή δυάδα C-G μετακινείται από την πρώτη θέση στην τρίτη, ενώ άλλες δυάδες φτάνουν στις πρώτες θέσεις, όπως η E-A που μετακινείται από την 13η στην 7η θέση. Αυτή η αλλαγή στις κατανομές είναι σαφώς ορατή στις τριάδες, όπου η πρώτη θέση (G-C-G) καταλαμβάνεται από την 11η (D-G-C) κατά τη μετάβαση από την κατάταξη συνολικά σε αυτή των ορίων των τμημάτων. Επίσης, η προηγούμενη τελευταία θέση γίνεται 4η, ενώ ένα μεγάλο μέρος των 20 κορυφαίων τριάδων συνολικά δεν κατατάσσεται στις 20 κορυφαίες μεταξύ των ορίων των τμημάτων.

Οι 20 κορυφαίες Δυάδες Συγχορδιών — Συνολικά έναντι Μεταξύ Τμημάτων

Θέση	Συνολικά (Δυάδες)	Πλήθος	Μεταξύ Τμημάτων (Δυάδες)	Πλήθος
1	C → G	1223564	G → C	122346
2	G → D	1014246	D → G	103336
3	G → C	980146	A → D	70224
4	D → G	852062	A → D	68636
5	F → C	729076	C → D	63256
6	D → A	637851	C → F	60017
7	A → D	965292	E → A	54259
8	C → F	521201	F → C	45411
9	A → E	483153	D → A	41526
10	G → Amin	468978	G → Amin	40044
11	C → D	447004	A → E	35774
12	F → G	434500	D → Emin	33164
13	E → A	424090	D → C	30465
14	Emin → C	385760	G → F	30377
15	D → Emin	382211	G → Emin	28707
16	Amin → G	379815	C → Amin	28310
17	D → C	377039	B → E	26214
18	Amin → F	376451	C → Emin	21673
19	G → F	365110	A → G	21601
20	C → Amin	346017	Amin → C	20189

Οι 20 κορυφαίες Τριάδες Συγχορδιών — Συνολικά έναντι Μεταξύ Τμημάτων

Θέση	Συνολικά (Τριάδες)	Πλήθος	Μεταξύ Τμημάτων (Τριάδες)	Πλήθος
1	G → C → D	354285	D → G → C	32600
2	C → G → D	306824	C → G → C	28711
3	C → G → C	298049	G → C → F	24613
4	D → G → D	274881		24430
5	F → C → G	263670	C → D → G	23931
6	G → D → G	250124	D → G → D	23594
7	C → F → C	216719	F → G → C	23482
8	C → G → Amin	209126	A → D → G	22613
9	G → D → A	196786	G → D → G	22461
10	A → D → A	190976	G → A → D	17267
11	D → G → C	190470	A → D → A	16085
12	C → D → G	190050	D → A → D	14861
13	G → D → Emin	182484	E → A → D	14668
14	F → G → C	178555	A → E → A	14590
15	Emin → C → G	176361	C → G → Amin	14506
16	D → A → D	173577	F → C → F	13780
17	Amin → F → C	169207	E → A → E	13068
18	D → C → G	164655	G → D → Emin	12484
19	G → C → F	160997	Amin → F → C	11859
20	F → C → F	159881	C → F → C	11566

Figure 1.3.13: Κορυφαία δυάδες (αριστερά) και τριάδες (δεξιά) σε όλες τις μεταβάσεις και στις μεταβάσεις στην αλλαγή τμημάτων.

Για να ερευνήσουμε περαιτέρω, στα Σχήματα 1.3.14 και 1.3.15 παρουσιάζουμε τις 10 πιο συνηθισμένες συγχορδίες και δυάδες και τις κατανομές των 5 κορυφαίων επόμενων συγχορδιών που επιλέχθηκαν σε αυτή την ακολουθία τόσο εντός κάθε μέρους όσο και κατά τις μεταβάσεις των τμημάτων. Οι κατανομές μεταξύ των τμημάτων τοποθετούν την εν λόγω συγχορδία ή δυάδα ως τα τελευταία στοιχεία ενός τμήματος και ελέγχουν ποια συγχορδία ακολουθεί, ως η πρώτη του επόμενου τμήματος.

Υπολογίζοντας την απόσταση συνολικής διακύμανσης (ΑΣΔ) 1.3.1, μπορούμε να δούμε ότι οι κατανομές εντός και κατά τις μεταβάσεις των τμημάτων διαφέρουν για τα διγράμματα κατά περίπου 52% (συνολικά) και 17% (σταθμισμένα ως προς τη συχνότητα) και για τις τριάδες κατά περίπου 57% και 23% αντίστοιχα. Ο τύπος της απόστασης συνολικής διακύμανσης μεταξύ δύο κατανομών φαίνεται παρακάτω:

$$\text{ΑΣΔ}(P,Q) = \frac{1}{2}\sum_{x\in\mathcal{X}}\big|P(x) - Q(x)\big| \tag{1.3.1}$$

όπου P και Q είναι κατανομές πιθανότητας για το ίδιο σύνολο γεγονότων $\mathcal{X}$, και x αντιπροσωπεύει κάθε πιθανό γεγονός.

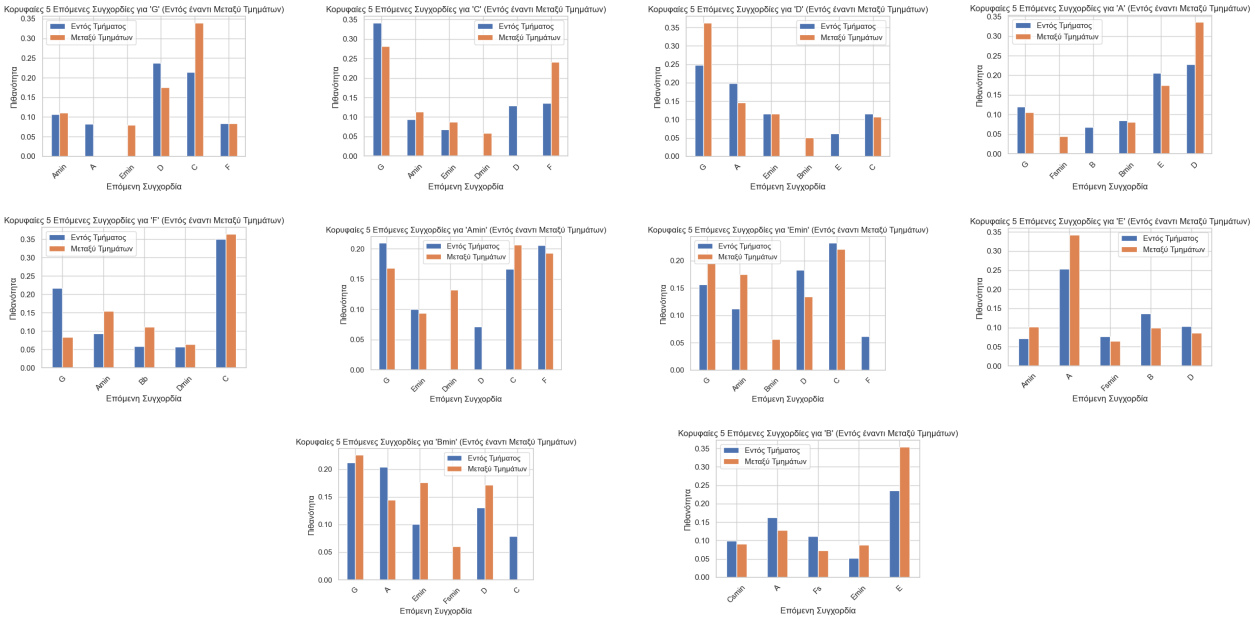


Figure 1.3.14: Οι 10 κορυφαίες συγχορδίες και πώς αλλάζουν οι κατανομές των επόμενων συγχορδιών τους κατά τη μετάβαση σε επόμενο μέρος.

Όσον αφορά τις δυάδες, αξιοσημείωτη περίπτωση αποτελεί η συγχορδία G, η οποία έχει τους πιο συνηθισμένους διαδόχους της εντός των μερών, τη συγχορδία D με περίπου 24% και τη συγχορδία C να ακολουθεί σε δεύτερη θέση με 22%, να ανταλλάσσουν θέσεις και τη συγχορδία C να κυριαρχεί με 34% έναντι του 17% της συγχορδίας D. Επιπλέον, ακολουθώντας τη συγχορδία $Amin$ από την εσωτερική κατανομή, μπορούμε να δούμε ότι οι συγχορδίες F και G είναι οι κορυφαίες υποψήφιες με περίπου 20%, με τη συγχορδία C να καταλαμβάνει την τρίτη θέση με 17%. Αυτή η δυναμική αλλάζει στα όρια των τμημάτων, με τη συγχορδία C να καταλαμβάνει την πρώτη θέση ανταλλάσσοντας με τη συγχορδία G, ενώ το ποσοστό της συγχορδίας F πέφτει κάτω από 19%. Άλλες ενδιαφέρουσες παρατηρήσεις είναι ότι στις περισσότερες περιπτώσεις, όπως οι συγχορδίες D, A, E και B, η πιο δημοφιλής επιλογή εντός των τμημάτων ενισχύεται στα όρια των τμημάτων. Στις υπόλοιπες περιπτώσεις, οι κυρίαρχες επιλογές παραμένουν οι ίδιες, ενώ η ιεραρχία κατανομής των λιγότερο επιδραστικών επιλογών ποικίλλει. Για παράδειγμα, μετά τη συγχορδία F, η συγχορδία C παραμένει η πιο κοινή επιλογή, αλλά η συγχορδία G μετακινείται από τη δεύτερη στην τέταρτη πιο κοινή, χάνοντας περίπου 16%, παρόμοια με τις συγχορδίες A και D που χάνουν περίπου 6% και γίνονται η τέταρτη επιλογή, αντί για τη δεύτερη, μετά τις συγχορδίες $Bmin$ και $Emin$, αντίστοιχα. Λαμβάνοντας υπόψη τη συγχορδία C, φαίνεται ότι η κατανομή στα όρια των τμημάτων μειώνει τη διαφορά μεταξύ των εμφανίσεων των συγχορδιών F και G στο 5%, από το 20% που είχαν στην κατανομή στο ίδιο τμήμα.

Συνεχίζοντας με τα μοτίβα τριάδων που φαίνονται στο Σχήμα 1.3.15, εντοπίζουμε μεγαλύτερες διακυμάνσεις μεταξύ των κατανομών εντός και στα όρια των τμημάτων. Οι πιο αξιοσημείωτες από αυτές είναι οι αλλαγές μετά την πρόοδο C-G, όπου η πιο διαδεδομένη επιλογή αλλάζει από D και C με 27% και 24% αντίστοιχα, στην κατανομή εντός, σε C με 34% και D με 13% στην κατανομή στα όρια. Με παρόμοιο τρόπο, οι πιο εμφανείς συγχορδίες μετά τις προόδους G-C, D-G, F-C και A-D αλλάζουν από G, D, G και A σε F, C, F και G, αυξάνοντας κατά 16%, 16%, 11% και 16% αντίστοιχα. Οι υπόλοιπες τριάδες δείχνουν μικρή διακύμανση στα ποσοστά των κυρίαρχων υποψήφιων συγχορδιών, ενώ οι άλλες προσαρμόζονται ασήμαντα.

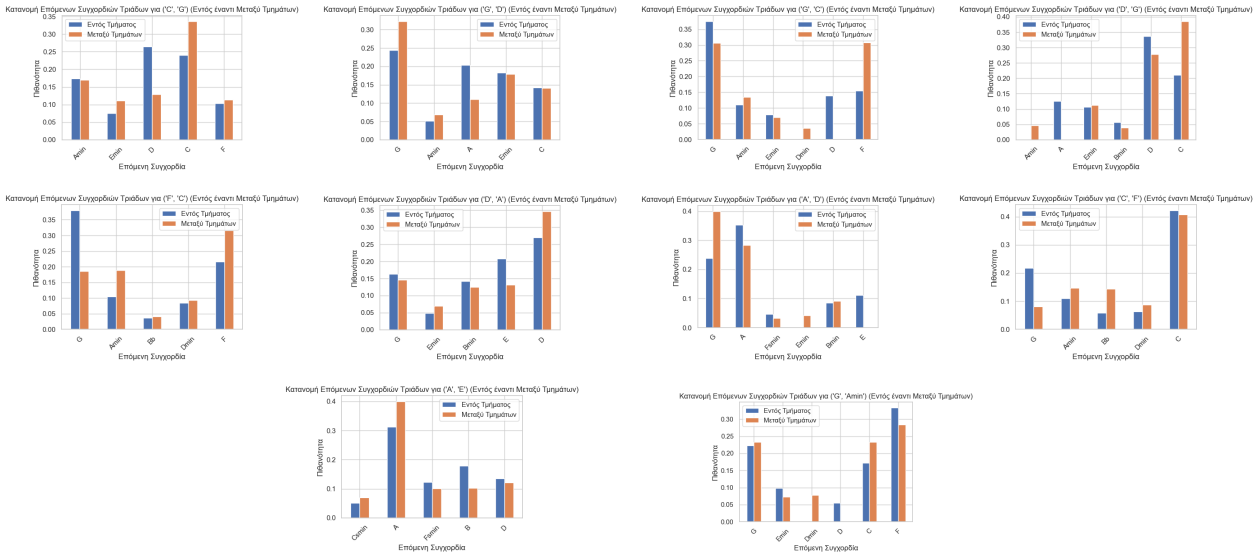


Figure 1.3.15: Οι 10 κορυφαίες δυάδες συγχορδιών και πώς αλλάζουν οι κατανομές των επόμενων συγχορδιών τους κατά τη μετάβαση σε επόμενο μέρος.

1.3.4 Ανάλυση μέσω Μοντέλου Λανθάνουσας Απόστασης

Αυτή η ενότητα περιγράφει την προεπεξεργασία του συνόλου δεδομένων και τις μεθόδους που εφαρμόστηκαν για τη δημιουργία μιας δομής τύπου γραφήματος από n-grams, με σκοπό τη μοντελοποίηση τραγουδιών χρησιμοποιώντας μοντέλα λανθάνουσας απόστασης σε σχέση με το είδος και τη δεκαετία κυκλοφορίας τους. Η βασική ιδέα είναι ότι μπορούμε να σχεδιάσουμε μια μέθοδο για να αναπαραστήσουμε κάθε τραγούδι σε έναν λανθάνοντα χώρο μέσω των n-grams του και να προσπαθήσουμε να δούμε αν αυτή η αναπαράσταση βοηθά σε εργασίες ταξινόμησης, όπως η ταξινόμηση ανά είδος και δεκαετία, με τα αποτελέσματα να εμφανίζονται στην ενότητα 1.4.

Εξισορρόπηση Συνόλου Δεδομένων

Για να διασφαλιστεί η ισότιμη εκπροσώπηση κάθε κατηγορίας, το πρώτο βήμα ήταν ο καθαρισμός και η εξισορρόπηση του ακατέργαστου συνόλου δεδομένων. Οι εγγραφές που δεν περιείχαν μεταδεδομένα σχετικά με το είδος ή τη δεκαετία απορρίφθηκαν, μαζί με εκείνες που περιείχαν μη έγκυρα σύμβολα συγχορδιών. Τα υπόλοιπα δεδομένα εξισορροπήθηκαν στη συνέχεια ανά κατηγορία, διατηρώντας τον ίδιο αριθμό δειγμάτων ανά έγκυρη ομάδα. Οι ομάδες με λίγα δείγματα, λιγότερα από 200, αποκλείστηκαν για να αποφευχθεί η υποεκπροσώπηση ορισμένων κατηγοριών. Το τελικό σύνολο δεδομένων αποτελείτο από 2.814 δείγματα ανά κατηγορία είδους και 1.748 δείγματα ανά κατηγορία δεκαετίας. Στη συνέχεια, πραγματοποιήθηκαν στρωματοποιημένες διασπάσεις εκπαίδευσης/ελέγχου χρησιμοποιώντας το 80% των δειγμάτων για εκπαίδευση και το 20% για έλεγχο, προκειμένου να διατηρηθεί η κατανομή των κατηγοριών και στις δύο διασπάσεις.

Επεξεργασία Συγχορδιών και Εξαγωγή Πλειάδων

Πριν από την κωδικοποίηση κάθε συγχορδίας, αυτές υποβλήθηκαν σε προεπεξεργασία για την τυποποίηση της αναπαράστασής τους. Οι ετικέτες των δομικών τμημάτων απορρίφθηκαν και διατηρήθηκε μόνο η ρίζα κάθε συγχορδίας, αφαιρώντας τις πληροφορίες σχετικά με την ποιότητα, τις επεκτάσεις και το μπάσο. Στη συνέχεια, κάθε ρίζα μετατράπηκε σε μια ατονική τάξη τόνου, χρησιμοποιώντας ακέραιους αριθμούς από 0 έως 11 για την αναπαράσταση κάθε συγχορδίας, καταγράφοντας τα δώδεκα ημιτόνια σε μια οκτάβα. Οι ατονικές αναπαραστάσεις βοηθούν στη μείωση της πολυπλοκότητας που προκύπτει από τις πλειάδες καθώς αυξάνεται το n (μήκος πλειάδας). Μετά την απλοποίηση κάθε ακολουθίας, συλλέχθηκαν επικαλυπτόμενες πλειάδες για $n = 4, 5$ και 6, με έμφαση στην καταγραφή τοπικών αρμονικών μοτίβων. Για κάθε πλειάδα μετρήθηκε η απόλυτη συχνότητά της σε όλα τα τραγούδια και η σχετική συχνότητά της σε κάθε κατηγορία.

Γράφημα Κατηγοριών

Πριν από την κατασκευή του πλήρους γραφήματος πλειάδων, δημιουργήθηκε ένα μικρότερο γράφημα όπου κάθε κόμβος αντιπροσώπευε μια κατηγορία και οι άκρες αντανακλούσαν τις σχέσεις μεταξύ τους. Ο σκοπός αυτού του γραφήματος ήταν η άμεση αρχικοποίηση των λανθανουσών παραμέτρων του πλήρους γραφήματος πλειάδων, αντί για τυχαία αρχικοποίηση, κάθε πλειάδα ξεκινά πιο κοντά στην πιο σχετική κατηγορία της. Οι ακμές αυτού του μικρότερου γραφήματος ορίστηκαν μέσω ομοιότητας Soft Jaccard των προφίλ των πλειάδων τους, όπου ο αριθμητής αθροίζει τις ελάχιστες ποσοστιαίες συνεισφορές των κοινών πλειάδων και ο παρονομαστής αθροίζει τη μέγιστη συνεισφορά σε όλες τις πλειάδες. Αυτή η μέτρηση καταγράφει τόσο την παρουσία όσο και τη σχετική σημασία των κοινών μοτίβων σε όλες τις κατηγορίες. Τέλος, οι ακμές κανονικοποιήθηκαν σε μια σταθερή κλίμακα, με αποτέλεσμα ένα βεβαρημένο, μη κατευθυνόμενο γράφημα που κωδικοποιεί τις ομοιότητες των κατηγοριών.

Γράφημα Πλειάδων

Για το γράφημα πλειάδων, κάθε κόμβος αντιπροσωπεύει μια μοναδική πλειάδα και οι ακμές δημιουργήθηκαν με βάση τους πλησιέστερους γείτονες σε κάθε κατηγορίας. Πιο συγκεκριμένα, εντός κάθε κατηγορίας, οι πλειάδες ταξινομήθηκαν κατά κανονικοποιημένη ποσοστιαία συνεισφορά και συνολικό αριθμό, και ζεύγη πλειάδων με γειτονικές θέσεις (διαφορές θέσης ≤ 25) συνδέθηκαν. Δημιουργήθηκε ένα ξεχωριστό σύνολο ακμών που συνδέει κάθε πλειάδα με την κατηγορία που αντιπροσωπεύει περισσότερο, δημιουργώντας ένα άλλο σταθμισμένο, μη κατευθυνόμενο γράφημα που θα επιτρέψει αργότερα στους κόμβους κατηγοριών να πάρουν τη θέση τους στον λανθάνοντα χώρο μεταξύ των πλειάδων.

Χρωματισμός και Αντιστοίχιση κόμβων

Για να βελτιωθεί η ερμηνευσιμότητα, σε κάθε κόμβο πλειάδας αποδόθηκε ένα χρώμα κατηγορίας που αντιστοιχεί στην κατηγορία στην οποία είχε τη μεγαλύτερη σχετική συνεισφορά. Αυτές οι πληροφορίες εξάγονται από τις συνδέσεις μεταξύ των πλειάδων και των κατηγοριών που περιγράφονται παραπάνω. Αυτή η χρωματική κωδικοποίηση παρέχει έναν τρόπο οπτικής αναγνώρισης της ομαδοποίησης εντός του λανθάνοντος χώρου και επισήμανσης των πλειάδων που συνδέονται στενά με συγκεκριμένες κατηγορίες.

Επιπλέον, σε κάθε κόμβο πλειάδας αποδόθηκε μια μέτρηση αντιστοίχισης για να ποσοτικοποιηθεί η κατανομή της σε όλες τις κατηγορίες. Υπολογίστηκε ένα μέτρο βασισμένο στην απόκλιση Kullback-Leibler 1.3.3 από τις ποσοστιαίες συνεισφορές της πλειάδας, καταγράφοντας τη σχετική της ειδικότητα σε συγκεκριμένες κατηγορίες. Αυτές οι αντιστοιχίσεις επιτρέπουν σε νέα, άγνωστα τραγούδια να συνδεθούν με το γράφημα πελιάδων, να τοποθετηθούν στον λανθάνοντα χώρο και να ερμηνευθούν με βάση την ομοιότητά τους με το υπάρχον σύνολο δεδομένων. Ο τύπος της μέτρησης απόκλισης Kullback-Leibler έχει ως εξής:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{c=1}^{C} p_c \log \frac{p_c}{q_c}, \tag{1.3.2}$$

όπου p_c είναι το παρατηρούμενο ποσοστό μιας πλειάδας στην κατηγορία c και q_c είναι μια κατανομή αναφοράς.

Στην εφαρμογή μας, χρησιμοποιούμε μια ομοιόμορφη κατανομή αναφοράς $q_c = \frac{1}{C}$, η οποία απλοποιεί τον τύπο σε:

$$\text{KL-εμπνευσμένη}(x) = \sum_{c=1}^{C} p_c \log p_c + \log C. \tag{1.3.3}$$

Αυτός ο τύπος μετρά πόσο συγκεντρωμένη είναι μια πλειάδα σε όλες τις κατηγορίες:

- Οι τιμές κοντά στο 0 υποδηλώνουν ότι η πλειάδα είναι ομοιόμορφα κατανεμημένη σε όλες τις κατηγορίες.
- Οι υψηλότερες τιμές υποδηλώνουν ότι η πλειάδα είναι πιο συγκεκριμένη για κάθε κατηγορία, συμβάλλοντας κυρίως σε ένα υποσύνολο κατηγοριών.

Χρησιμοποιώντας αυτή τη μετρική, αξιολογούμε πόση πληροφορία συνεισφέρει κάθε πλειάδα στον λανθάνοντα χώρο και πόσο έντονα η ύπαρξή της σε ένα τραγούδι επηρεάζει τον τρόπο με τον οποίο πρέπει να κατηγοριοποιηθεί.

Μοντελοποίηση με Λανθάνουσα Απόσταση

Αφού προσδιορίστηκαν οι σχέσεις μεταξύ κατηγοριών, μεταξύ των πλειάδων αλλά και οι σχέσεις μεταξύ των πλειάδων με τις κατηγοριών, το μοντέλο λανθάνουσας απόστασης εκπαιδεύτηκε ώστε να ενσωματώνει τόσο τις κατηγορίες όσο και τις πλειάδες σε έναν κοινό λανθάνοντα χώρο. Η διαδικασία εκπαίδευσης διαρθρώθηκε σε τρεις διαδοχικές φάσεις για την παραγωγή ερμηνεύσιμων και σημασιολογικά σημαντικών ενσωματώσεων. Τα μοντέλα υλοποιήθηκαν σε Python και εκπαιδεύτηκαν χρησιμοποιώντας PyTorch με τον βελτιστοποιητή Adam. Η εκπαίδευση διήρκεσε 100 εποχές με 100 βήματα ανά εποχή, χρησιμοποιώντας ρυθμό μάθησης 0,01 και μεγέθη παρτίδων 4.000 για τις πλειάδες και τον αριθμό των κατηγοριών για τα γραφήματα κατηγοριών. Είναι σημαντικό να σημειωθεί ότι σε κάθε βήμα κάθε εποχής, το μοντέλο εκπαίδευσε μόνο τις θέσεις των επιλεγμένων πλειάδων, που καθορίστηκαν από το μέγεθος της παρτίδας, μεταξύ τους. Η υλοποίηση βασίζεται στο μοντέλο λανθάνουσας απόστασης που είναι διαθέσιμο στο github [15], το οποίο χρησιμοποιεί μια προσαρμοσμένη αρνητική απώλεια log-likelihood, που υπολογίζεται από τις αποστάσεις ζευγών στον λανθάνοντα χώρο.

Φάση 1: Εκπαίδευση του Γραφήματος Κατηγοριών

Η πρώτη φάση επικεντρώνεται μόνο στους κόμβους κατηγοριών. Η εκπαίδευση του γραφήματος κατηγοριών τοποθέτησε κάθε κόμβο κατηγορίας στον λανθάνοντα χώρο πιο κοντά στους κόμβους κατηγοριών που ήταν πιο σχετικοί, όπως φαίνεται στο Σχήμα 1.3.16.

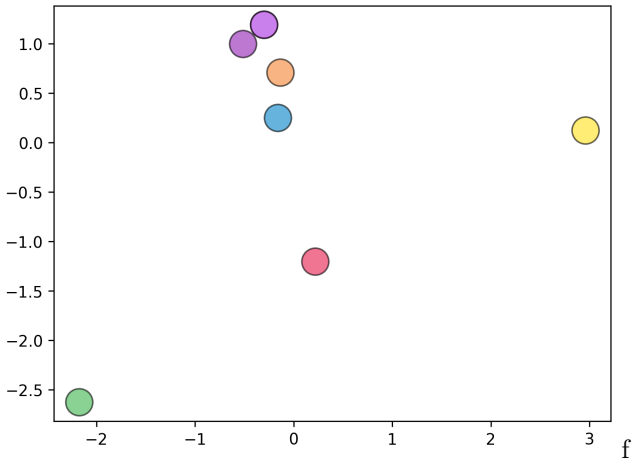


Figure 1.3.16: Ένα παράδειγμα κόμβων κατηγοριών μετά την πρώτη φάση εκπαίδευσης.

Φάση 2: Εκπαίδευση του Γραφήματος Πλειάδων

Η δεύτερη φάση αξιοποιεί τις λανθάνουσες θέσεις των κατηγοριών που υπολογίστηκαν στην πρώτη φάση για να αρχικοποιήσει τις λανθάνουσες θέσεις των πλειάδων. Κάθε κόμβος αρχικοποιείται στη θέση του κόμβου κατηγορίας με τον οποίο συνδέεται περισσότερο, ενώ παράλληλα προστίθεται και κάποιος μικρός θόρυβος. Η εκπαίδευση του γραφήματος πλειάδων έχει ως αποτέλεσμα τις τελικές θέσεις τους στον λανθάνοντα χώρο, όπως φαίνεται στο Σχήμα 1.3.17.

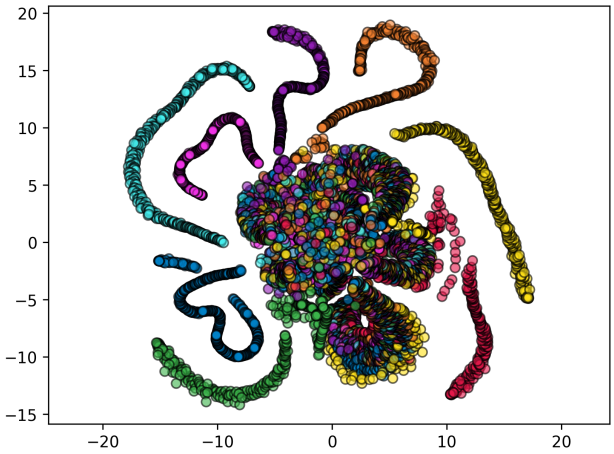


Figure 1.3.17: Ένα παράδειγμα κόμβων πλειάδων μετά τη δεύτερη φάση εκπαίδευσης.

Φάση 3: Επανατοποθέτηση των Κατηγοριών

Αφού οι πλειάδες έχουν φτάσει στις βελτιστοποιημένες θέσεις τους στον λανθάνοντα χώρο, οι κόμβοι κατηγοριών επανατοποθετούνται με βάση τις συνδέσεις τους με αυτές. Όπως αναφέρθηκε προηγουμένως, κάθε πλειάδα συνδέεται με μία μόνο κατηγορία, για τον χρωματισμό της, επιτρέποντας τον υπολογισμό της τελικής θέσης κάθε κατηγορίας ως σταθμισμένο κέντρο βάρους των συνδεδεμένων πλειάδων της, όπως φαίνεται στο Σχήμα 1.3.18.

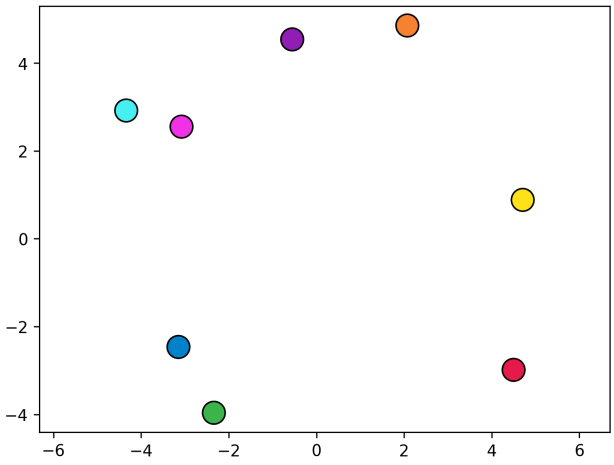


Figure 1.3.18: Ένα παράδειγμα κόμβων κατηγοριών μετά την τρίτη φάση εκπαίδευσης.

Το αποτέλεσμα των παραπάνω φάσεων είναι ένας λανθάνων χώρος όπου οι πλειάδες και οι κατηγορίες καταλαμβάνουν βελτιστοποιημένες θέσεις, αποκαλύπτοντας ερμηνεύσιμες γεωμετρικές σχέσεις που αντικατοπτρίζουν τις στατιστικές και σημασιολογικές συσχετίσεις τους. Παρακάτω, στα Σχήματα 1.3.19 και 1.3.20, μπορούν να παρατηρηθούν οι λανθάνουσες θέσεις των τετράδων, πεντάδων και εξάδων, καθώς και οι λανθάνουσες θέσεις των κατηγοριών είδους και δεκαετίας αντίστοιχα.

Από τα διαγράμματα είναι προφανές ότι σχηματίζονται ισχυρότερες, πιο ορατές συστάδες καθώς προχωράμε από τις τετράδες στις εξάδες. Οι πυκνές περιοχές που βρίσκονται πιο κοντά στο κέντρο αντιπροσωπεύουν περίπου το 90% του συνόλου των σημείων. Στην περίπτωση των ειδών, έχουμε συνολικά 11.117 μοναδικές τετράδες, 29.730 μοναδικές πεντάδες και 45.462 μοναδικές εξάδες. Όσον αφορά τις δεκαετίες, μετράμε 7.490 μοναδικές τετράδες, 16.436 μοναδικές πεντάδες και 21.289 μοναδικές εξάδες.

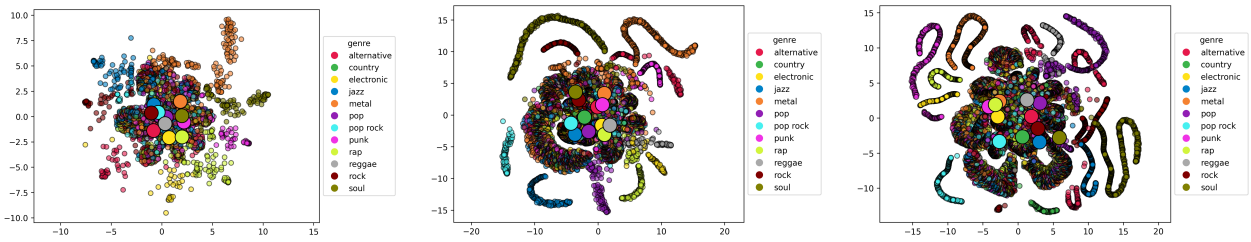


Figure 1.3.19: Οι λανθάνουσες θέσεις των τετράδων, πεντάδων και εξάδων, από αριστερά προς τα δεξιά, με βάση τις συνδέσεις που δημιουργήθηκαν από την ομοιότητα των ειδών.

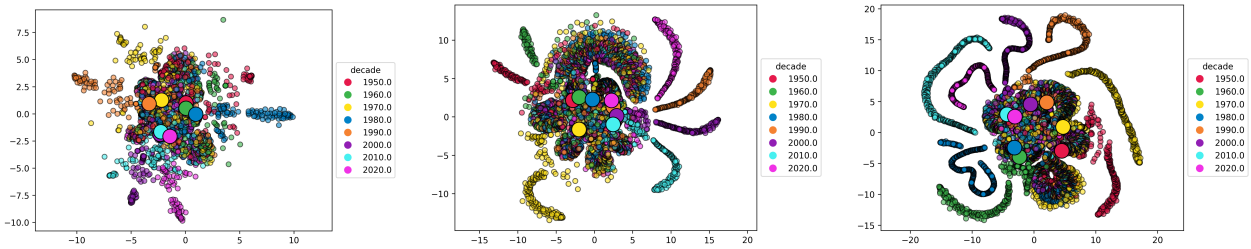


Figure 1.3.20: Οι λανθάνουσες θέσεις των τετράδων, πεντάδων και εξάδων, από αριστερά προς τα δεξιά, με βάση τις συνδέσεις που δημιουργήθηκαν από την ομοιότητα των δεκαετιών.

Εστιάζοντας στα διαγράμματα των πεντάδων και εξάδων, όπου μπορούμε να διακρίνουμε καλύτερα τη διαφοροποίηση των κατηγοριών, παρατηρούμε ορισμένα μοτίβα. Τα πιο αξιοσημείωτα είναι τα είδη Πάνκ και Μέταλ, τα οποία βρίσκονται κοντά και στις δύο περιπτώσεις, ενώ το ίδιο ισχύει και για τη Τζαζ και την Κάντρι. Επιπλέον, η Ροκ και η Σόουλ φαίνεται να έχουν παρόμοιες προτιμήσεις πλειάδων, όπως και η Ρέγκε και η Ποπ. Όσον αφορά τις δεκαετίες, μπορούμε να δούμε καθαρά ότι οι δεκαετίες του 1960 και του 1980 βρίσκονται κοντά και ότι οι δεκαετίες του 2000, του 2010 και του 2020 είναι δίπλα η μία στην άλλη τόσο στα διαγράμματα πεντάδων όσο και στα διαγράμματα εξάδων, ενώ η δεκαετία του 1990 βρίσκεται σε κοντινή απόσταση. Η δεκαετία του 1970 φαίνεται να απομακρύνεται από τις δύο ομάδες που σχηματίστηκαν.

1.4 Πειράματα

Σε αυτήν την ενότητα παρουσιάζουμε τα πειράματα που διεξήχθησαν για την αξιολόγηση του προτεινόμενου πλαισίου μας. Τα πειράματα χωρίζονται σε δύο κύριες κατηγορίες: ταξινόμηση και πρόβλεψη συγχορδιών. Όλες οι υλοποιήσεις έγιναν σε Python, χρησιμοποιώντας τη βιβλιοθήκη PyTorch για την κατασκευή και την εκπαίδευση του μοντέλου. Για τη βελτιστοποίηση, χρησιμοποιήσαμε τον βελτιστοποιητή Adam, ο οποίος παρέχει προσαρμοστικούς ρυθμούς μάθησης και αποτελεσματική σύγκλιση τόσο για τα αναδρομικά μοντέλα όσο και για τα μοντέλα λανθάνουσας απόστασης. Το μοντέλο λανθάνουσας απόστασης χρησιμοποίησε μια αρνητική συνάρτηση απώλειας log-likelihood, όπως περιγράφεται στο 1.3.4, ενώ τα αναδρομικά μοντέλα χρησιμοποίησαν απώλεια cross-entropy.

1.4.1 Ταξινόμηση ανά Είδος και Δεκαετία

Σε αυτή την ενότητα, περιγράφουμε κυρίως τη διαδικασία για την εκπαίδευση των ακολουθιακών μοντέλων, καθώς η διαδικασία μοντελοποίησης της λανθάνουσας απόστασης για την προεπεξεργασία και την εκπαίδευση περιγράφηκε στην ενότητα 1.3.4. Στα ακολουθιακά μοντέλα περιλαμβάνουμε τα αναδρομικά μοντέλα και την αρχιτεκτονική μοντέλου κατάστασης-χώρου του Mamba. Οι μέθοδοι αξιολόγησης περιγράφονται και για τις δύο αρχιτεκτονικές.

Αναπαράσταση και Προεπεξεργασία Δεδομένων

Για τα ακολουθιακά μοντέλα, χρησιμοποιήθηκαν μέτρα παρόμοια με αυτά που χρησιμοποιήθηκαν για τα μοντέλα λανθάνοντος χώρου, με σκοπό την προεπεξεργασία των ακατέργαστων δεδομένων και την προετοιμασία τους για εκπαίδευση. Το σύνολο δεδομένων υποβλήθηκε σε στάδιο εξισορρόπησης και καθαρισμού και για τις δύο κατηγορίες ταξινόμησης, και κάθε συγχορδία μετατράπηκε σε ακέραιο αριθμό που αντιστοιχούσε στην τάξη τόνου της ρίζας 0-11. Σε αντίθεση με τη διαδικασία του μοντέλου λανθάνουσας απόστασης, το σύνολο δεδομένων δεν χωρίστηκε σε ακολουθίες μήκους n αλλά για την εκπαίδευση χρησιμοποιήθηκαν ολόκληρες ακολουθίες τραγουδιών.

Αρχιτεκτονική Μοντέλων

Εξετάστηκαν τέσσερις τύποι ακολουθιακών αρχιτεκτονικών: τα απλά αναδρομικά νευρωνικά δίκτυα , τα δίκτυα αναδρομικών μονάδων με πύλες, τα δίκτυα μακροπρόθεσμης βραχυπρόθεσμης μνήμης και το μοντέλο Mamba. Κάθε μοντέλο έλαβε ως είσοδο ακολουθίες ακέραιων αριθμών συγχορδιών (0-11), τις επεξεργάστηκε μέσω ενός στρώματος ενσωμάτωσης για να μάθει σημαντικές αναπαραστάσεις κάθε συγχορδίας και παρήγαγε μια κατανομή πιθανότητας ανά κατηγορία μέσω ενός πλήρως συνδεδεμένου στρώματος εξόδου με ενεργοποίηση softmax για κατηγοριακή ταξινόμηση. Το κύριο μοντέλο αποτελείται από ένα στρώμα των κύριων διαδοχικών μονάδων υπολογισμού.

Εκπαίδευση

Τα μοντέλα εκπαιδεύτηκαν χρησιμοποιώντας τη συνάρτηση απώλειας cross-entropy και οι υπερπαραμέτροι ρυθμίστηκαν εμπειρικά για βέλτιστη απόδοση. Η διάσταση ενσωμάτωσης που χρησιμοποιήθηκε ήταν 4 και η κρυφή διάσταση του μοντέλου ήταν 256, η εκπαίδευση διήρκεσε έως και 100 εποχές, χρησιμοποιώντας μέγεθος παρτίδας 256, με πρόωρη διακοπή και προγραμματισμό ρυθμού μάθησης ξεκινώντας από 0,001 για ομαλότερη εκπαίδευση. Τα μοντέλα Mamba περιλαμβάνουν ορισμένες επιπλέον υπερπαραμέτρους. Χρησιμοποιήσαμε διάνυσμα κατάστασης μεγέθους 32, μέγεθος πυρήνα συνέλιξης 4 και επέκταση του μοντέλου μεγέθους 2.

Το σύνολο δεδομένων χωρίστηκε σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου (80/10/10). Το σύνολο επικύρωσης χρησιμοποιήθηκε για την παρακολούθηση της απόδοσης του μοντέλου κατά τη διάρκεια της εκπαίδευσης, με την κατάσταση του μοντέλου που πέτυχε τη χαμηλότερη απώλεια επικύρωσης να αποθηκεύεται και να χρησιμοποιείται αργότερα για την αξιολόγηση στο σύνολο ελέγχου.

Αξιολόγηση

Τόσο για τα μοντέλα λανθάνουσας απόστασης όσο και για τα ακολουθιακά μοντέλα, η απόδοση της ταξινόμησης αξιολογήθηκε χρησιμοποιώντας δείκτη ευστοχίας και ανάλυση σύγχυσης στο σύνολο ελέγχου, εντοπίζοντας κοινές λανθασμένες ταξινομήσεις μεταξύ των κατηγοριών.

Για την αξιολόγηση των μοντέλων που χρησιμοποιούν ανάδραση, καταγράφηκε επίσης η απώλεια ελέγχου για να συγκριθεί η απόδοση των μοντέλων. Για τα μοντέλα λανθάνουσας απόστασης, η προβλεπόμενη κατηγορία για κάθε τραγούδι προέκυψε υπολογίζοντας τις αποστάσεις στον λανθάνοντα χώρο από τις θέσεις των κόμβων κατηγορίας που προέκυψαν κατά τη διάρκεια της εκπαίδευσης και επιλέγοντας το ελάχιστο. Για τις αναδρομικές μεθόδους, η πρόβλεψη προέκυψε από την μέγιστη πιθανότητα που παρήγαγε το στρώμα softmax.

1.4.2 Πρόβλεψη Συγχορδιών

Το πρόβλημα πρόβλεψης συγχορδιών βασίζεται σε συμβολικές ακολουθίες συγχορδιών από το σύνολο δεδομένων Chordonomicon. Τα πειράματα σχεδιάστηκαν με σκοπό να αξιολογήσουν τον τρόπο με τον οποίο το μέγεθος του συνόλου εκπαίδευσης και το μήκος της ακολουθίας επηρεάζουν την απόδοση του μοντέλου. Κάθε τραγούδι στο σύνολο δεδομένων αντιπροσωπεύεται από μια ακολουθία συγχορδιών που συνοδεύεται από ετικέτες δομικών τμημάτων, όταν υπάρχουν τέτοιες σημειώσεις (<Verse_1>, <Chorus_2>). Για την προετοιμασία των δεδομένων για τη μοντελοποίηση ακολουθιών, η ακατέργαστη μορφή τους έπρεπε να υποβληθεί σε προεπεξεργασία, προκειμένου να παραχθούν οι δύο διαφορετικές αναπαραστάσεις που χρησιμοποιήσαμε για τα πειράματα.

Αναπαράσταση και Προεπεξεργασία Δεδομένων

Το ακατέργαστο σύνολο δεδομένων φιλτράρεται για να αφαιρεθούν οι ελλιπείς ή θορυβώδεις καταχωρήσεις. Προτιμήθηκαν τα τραγούδια με τουλάχιστον μεταδεδομένα για το είδος ή τη δεκαετία, ενώ αυτά που περιείχαν περίεργους συμβολισμούς συγχορδιών (με ενδείξεις αναστροφής χωρίς να αναφέρουν τη νότα του μπάσου «/ ») απορρίφθηκαν.

Κανονικοποίηση και Καθαρισμός Το πρώτο βήμα ήταν να μετατρέψουμε τις ετικέτες των μερών, όπως <Verse_1>, <Chorus_2>, σε κανονικές μορφές Verse, Chorus, αφαιρώντας τις αγκύλες και τα αριθμητικά επιθήματα. Αυτές οι ετικέτες διατηρήθηκαν μόνο για τη δημιουργία διαδοχών συγχορδιών ανά τμήμα και εξαιρέθηκαν από το κύριο λεξιλόγιο συγχορδιών.

Κατασκευή και Κωδικοποίηση Λεξιλογίου Στα πειράματα δοκιμάσαμε δύο διαφορετικές αναπαραστάσεις των συγχορδιών, η μία αντιμετωπίζοντας κάθε μοναδική συγχορδία ως έναν όρο και η άλλη αποσυνθέτοντας τη συγχορδία σε τρία μέρη: Ρίζα, Ποιότητα + Επεκτάσεις και Μπάσο.

- **Αναπαράσταση με 1-όρο:** Κάθε μοναδικό σύμβολο συγχορδίας (π.χ. `Cmaj7`, `Am`, `G/B`) αντιμετωπίστηκε ως ατομικός όρος. Αυτή η προσέγγιση καταγράφει πλήρεις αρμονικές πληροφορίες σε μία μόνο μονάδα, με αποτέλεσμα μια συμπαγή αναπαράσταση της ακολουθίας.
- **Αναπαράσταση με 3-όρους:** Κάθε συγχορδία αποσυντέθηκε σε τρία συστατικά:
  - **Ρίζα:** η τάξη του τόνου της συγχορδίας (π.χ. C, D#, F#);
  - **Ποιότητα + Επεκτάσεις:** ο τύπος της συγχορδίας και οι πρόσθετοι τόνοι (π.χ. maj7, min, sus4, add13);
  - **Μπάσο:** η νότα του μπάσου όταν καθορίζεται (π.χ., το E στο `C/E`), ή `N` για καμία.

  Κάθε συστατικό κωδικοποιήθηκε ως ένας ακέραιος αναγνωριστικός κωδικός και οι αντιστοιχίσεις μεταξύ των αναγνωριστικών κωδικών των συστατικών και των πλήρων αναγνωριστικών κωδικών των συγχορδιών αποθηκεύτηκαν για αποκωδικοποίηση.

Κατασκευή Συνόλων Δεδομένων Μετά το φιλτράρισμα των τραγουδιών χωρίς μεταδεδομένα, δημιουργήθηκαν διαδοχικά τρία σύνολα δεδομένων:

- **Φιλτραρισμένο σύνολο δεδομένων:** Περιέχει όλα τα τραγούδια με τουλάχιστον ένα πεδίο μεταδεδομένων (είδος ή δεκαετία), με τις δομικές ετικέτες να έχουν αφαιρεθεί. Κατάλληλο για αξιολόγηση ολόκληρων τραγουδιών.
- **Σύνολο δεδομένων με ετικέτες τραγουδιών:** Υποσύνολο του φιλτραρισμένου συνόλου δεδομένων που περιέχει μόνο τραγούδια με δομικές ετικέτες. Στη συνέχεια, τα τραγούδια με ετικέτες χρησιμοποιούνται για τη δημιουργία του τμηματοποιημένου συνόλου δεδομένων και αργότερα διατηρούνται χωρίς ετικέτες για αξιολόγηση.
- **Τμηματοποιημένο σύνολο δεδομένων:** Κάθε καταχώριση αντιστοιχεί σε ένα μεμονωμένο τμηματοποιημένο τμήμα που εξάγεται από το σύνολο δεδομένων με ετικέτες τραγουδιών (π.χ. μια στροφή ή ένα ρεφρέν). Οι ετικέτες και οι σχετικές διαδοχές συγχορδιών διατηρούνται για μοντελοποίηση και αξιολόγηση σε επίπεδο τμήματος.

Όλα τα μοντέλα εκπαιδεύτηκαν στο τμηματοποιημένο σύνολο δεδομένων, ενώ η αξιολόγηση πραγματοποιήθηκε στο φιλτραρισμένο σύνολο δεδομένων, στο σύνολο δεδομένων με ετικέτες τραγουδιών και στο τμηματοποιημένο σύνολο δεδομένων μετά το φιλτράρισμα των τραγουδιών που είχαν τουλάχιστον ένα μέρος που ανήκε στο σύνολο δεδομένων εκπαίδευσης ή επικύρωσης.

Αρχιτεκτονική Μοντέλων

Για τα προβλήματα εξετάστηκαν τέσσερις αρχιτεκτονικές: τα απλά αναδρομικά νευρωνικά δίκτυα, οι αναδρομικές μονάδες με πύλες, τα δίκτυα μακράς και βραχείας μνήμης και το Mamba. Τα μοντέλα λαμβάνουν ακολουθίες είτε κωδικοποιημένων συγχορδιών (1-όρος) είτε κωδικοποιημένων συστατικών συγχορδιών (3-όροι) και χρησιμοποιούν ένα επίπεδο ενσωμάτωσης για να δημιουργήσουν σημαντικές ενσωματώσεις που θα υποβληθούν σε

επεξεργασία από την κύρια μονάδα. Η έξοδος της μονάδας συλλέγεται στη συνέχεια από πλήρως συνδεδεμένα στρώματα, με μία κεφαλή ανά συγχορδία ή 3 κεφαλές, μία ανά συστατικό συγχορδίας. Για την αναπαράσταση 3-όρους, οι κεφαλές των συστατικών συνδυάζονται στη συνέχεια για να προσδιοριστεί ποια συγχορδία από το γνωστό λεξιλόγιο ταιριάζει καλύτερα. Η προσέγγιση πολλαπλών κεφαλών επιτρέπει στο μοντέλο να αξιοποιεί ταυτόχρονα την εποπτεία σε επίπεδο συστατικών και πλήρους συγχορδίας.

Εκπαίδευση

Τα μοντέλα εκπαιδεύτηκαν χρησιμοποιώντας απώλεια cross-entropy, με τη συνολική απώλεια της προσέγγισης 3-όρων να είναι το άθροισμα των απωλειών των συστατικών και των πλήρων συγχορδιών. Οι υπερπαραμέτροι ήταν οι εξής: κρυφή διάσταση 256, ένα κύριο επίπεδο μονάδας, μέγεθος παρτίδας 4096, ρυθμός μάθησης 0,005, μέγιστο 200 εποχές και διάσταση ενσωμάτωσης 128 για τις συγχορδίες και 16 για κάθε συνιστώσα.

Για τη βελτίωση της απόδοσης και της γενίκευσης, χρησιμοποιήθηκαν πρόσθετοι μηχανισμοί. Χρησιμοποιήθηκε πρόωρη διακοπή σε συνδυασμό με προγραμματισμό ρυθμού μάθησης για την αποφυγή υπερπροσαρμογής, ενώ μια στρωματοποιημένη διαίρεση εξασφάλισε ότι τα σύνολα επικύρωσης και ελέγχου της εκπαίδευσης δεν είχαν επικαλυπτόμενα τραγούδια και διατήρησαν τις κατανομές ετικετών. Επίσης, κατά τη διάρκεια όλων των φάσεων, για την αύξηση της ανθεκτικότητας σε ακολουθίες μεταβλητού μήκους, οι ακολουθίες κόβονταν τυχαία κατά τη δειγματοληψία, εξασφαλίζοντας παράλληλα ότι η στοχευόμενη συγχορδία προς πρόβλεψη ακολουθούσε τουλάχιστον 4 συγχορδίες.

Αξιολόγηση

Και για τα δύο πειράματα, η μέση ακρίβεια και η μέση απώλεια χρησιμοποιήθηκαν για τη μέτρηση της απόδοσης σε πολλαπλές εκτελέσεις. Για να εμπλουτιστεί η αξιολόγηση, πραγματοποιήθηκε ανάλυση σύγχυσης σχετικά με τις ασυμφωνίες, προκειμένου να μετρηθεί η επίδρασή τους στην απόδοση του μοντέλου.

1.4.3 Αποτελέσματα

Σε αυτή την ενότητα παρουσιάζουμε τα αποτελέσματα των πειραμάτων μας. Το πρώτο μέρος επικεντρώνεται στο πρόβλημα της ταξινόμησης, ενώ το δεύτερο μέρος ασχολείται με το πρόβλημα της πρόβλεψης συγχορδιών.

Ταξινόμηση ανά Είδος και Δεκαετία

Ο πίνακας 1.2 δείχνει την ακρίβεια και την απώλεια των εκπαιδευμένων μοντέλων τόσο για τις εργασίες ταξινόμησης είδους όσο και δεκαετίας. Τα αποτελέσματα μπορούν να ομαδοποιηθούν σε δύο κατηγορίες. Στην πρώτη ομάδα, το μοντέλο λανθάνουσας απόστασης και το απλό αναδρομικό νευρωνικό δίκτυο επιτυγχάνουν παρόμοια αποτελέσματα, ξεπερνώντας ελάχιστα την τυχαία εικασία που είναι 0,125 για τις οκτώ δεκαετίες και 0,08 για τα δώδεκα είδη. Η δεύτερη ομάδα, που περιλαμβάνει την αναδρομική μονάδα με πύλες, το δίκτυο μακράς-βραχείας μνήμης και τις αρχιτεκτονικές Mamba, επιτυγχάνει σημαντικά καλύτερη απόδοση, σχεδόν διπλασιάζοντας την ακρίβεια του μοντέλου λανθάνουσας απόστασης στο πρόβλημα ταξινόμησης είδους και βελτιώνοντας την ακρίβεια ταξινόμησης δεκαετίας κατά περίπου 10%. Αξίζει να σημειωθεί ότι η αναδρομική μονάδα με πύλες παρουσιάζει χαμηλότερες τιμές απώλειας από τα άλλα μοντέλα, ακόμη και αν δεν είναι το μοντέλο με την καλύτερη απόδοση όσον αφορά την ακρίβεια στο πείραμα δεκαετίας.

Για να εξετάσουμε περαιτέρω την απόδοση του μοντέλου, παρουσιάζουμε χάρτες θερμότητας σύγχυσης στο Σχήμα 1.4.1. Οι χάρτες θερμότητας δημιουργήθηκαν συνδυάζοντας τις αναντιστοιχίες αξιολόγησης από όλα τα μοντέλα με τον συνδυασμένο πίνακα που δείχνει συμφωνία άνω του 90% με τον πίνακα σύγχυσης κάθε μεμονωμένου μοντέλου.

Στον χάρτη θερμότητας των ειδών, ορισμένα ζεύγη ειδών παρουσιάζουν μεγαλύτερη ομοιότητα από άλλα. Η Τζαζ συγχέεται συχνότερα με την Κάντρι, ενώ παρουσιάζει επίσης πολλαπλές συγχύσεις με τη σόουλ. Επιπλέον, η Πανκ συχνά συγχέεται με τη Μέταλ. Συνολικά, τα περισσότερα σφάλματα των μοντέλων αφορούν την εσφαλμένη ταξινόμηση άλλων ειδών με την Κάντρι.

Στον χάρτη θερμότητας των δεκαετιών, η δεκαετία του 1960 παρουσιάζει μεγάλη ομοιότητα με τη δεκαετία του 1950, με τη δεκαετία του 1950 να είναι η πιο συνηθισμένη εσφαλμένη ταξινόμηση. Μια άλλη μικρή ομάδα παρατηρείται μεταξύ των δεκαετιών 2000, 2010 και 2020.

Table 1.2: Ακρίβεια και απώλεια για κάθε μοντέλο ομαδοποιημένα ανά κατηγορία.

Κατηγορία	**Μοντέλο**	**Ακρίβεια**	**Απώλεια**
Είδος	LDM	0.0917	-
	RNN	0.1190	2.4610
	LSTM	0.1670	2.3915
	GRU	**0.1889**	**2.3608**
	Mamba	0.1753*	2.3905*
Δεκαετία	LDM	0.1326	-
	RNN	0.1344	2.0814
	LSTM	**0.2259**	1.9623*
	GRU	0.2252*	**1.9557**
	Mamba	0.2223	1.9831

Τα καλύτερα αποτελέσματα είναι σε **έντονη γραφή**, τα δεύτερα καλύτερα είναι σημειωμένα με *.

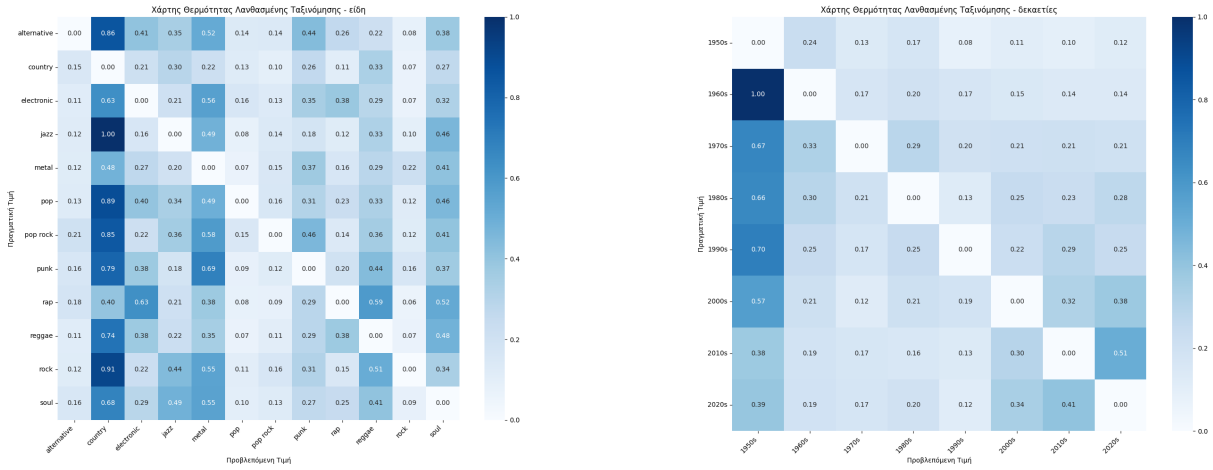
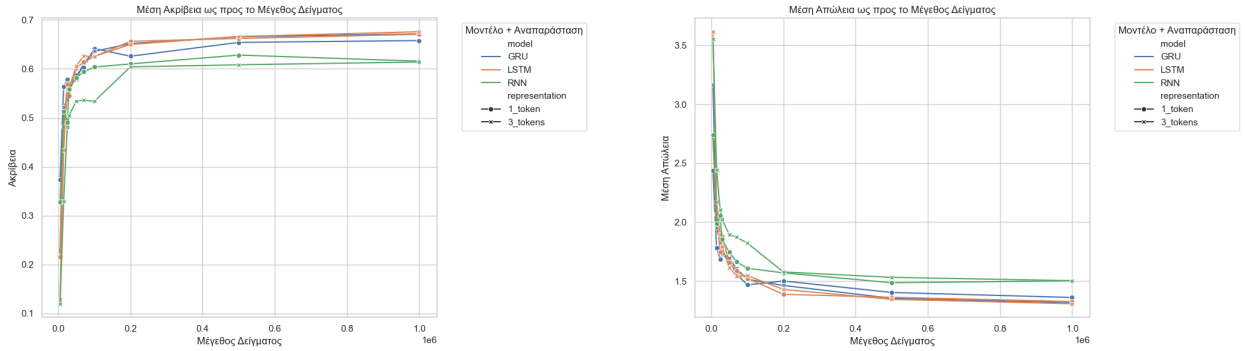


Figure 1.4.1: Χάρτες θερμότητας της σύγχυσης για το πρόβλημα της ταξινόμησης ανά είδος (αριστερά) και το πρόβληημα ταξινόμησης ανά δεκαετία (δεξιά).

Πρόβλεψη Συγχορδιών

Τα πειράματα πρόβλεψης συγχορδιών αποτελούνται από δύο κύριες αναλύσεις. Πρώτον, ερευνήσαμε πώς το μέγεθος του συνόλου δεδομένων εκπαίδευσης επηρεάζει την απόδοση του μοντέλου, όπως φαίνεται στο Σχήμα 1.4.2. Η απόδοση βελτιώνεται δραματικά καθώς το μέγεθος του δείγματος αυξάνεται από μερικές χιλιάδες εγγραφές (συγκεκριμένα 5.000) σε 200.000, με μόνο μικρές βελτιώσεις πέραν αυτού έως 1.000.000 δείγματα. Τα απλά αναδρομικά νευρωνικά δίκτυα έχουν σταθερά χειρότερη απόδοση από τα άλλα μοντέλα, ενώ η αναπαράσταση 3-όρων έχει ελαφρώς καλύτερη απόδοση από την αναπαράσταση 1 token.

Ο πίνακας 1.3 συνοψίζει τα μέσα αποτελέσματα από πολλαπλές εκτελέσεις χρησιμοποιώντας το μέγεθος δείγματος με την καλύτερη απόδοση (1.000.000 δείγματα). Όταν δοκιμάστηκε με ολόκληρες ακολουθίες τραγουδιών που έχουν ετικέτες μερών, η ακρίβεια μειώνεται και η απώλεια αυξάνεται. Η απόδοση μειώνεται περαιτέρω όταν δοκιμάζεται σε σύνολα δεδομένων που δεν έχουν ετικέτες δομής. Η αναπαράσταση με 3-όρους γενικά υπερτερεί της αναπαράστασης 1-όρου, εκτός από τα απλά αναδρομικά νευρωνικά μοντέλα. Όσον αφορά την ακρίβεια, το μοντέλο μακράς-βραχείας μνήμης επιτυγχάνει τα καλύτερα αποτελέσματα, ενώ τα δίκτυα αναδρομικών μονάδων με πύλες παρουσιάζουν ελαφρώς καλύτερη απόδοση στην απώλεια, παρά τη μικρή διαφορά στην ακρίβεια.



Σωρευτική Ακρίβεια | Σωρευτική Απώλεια

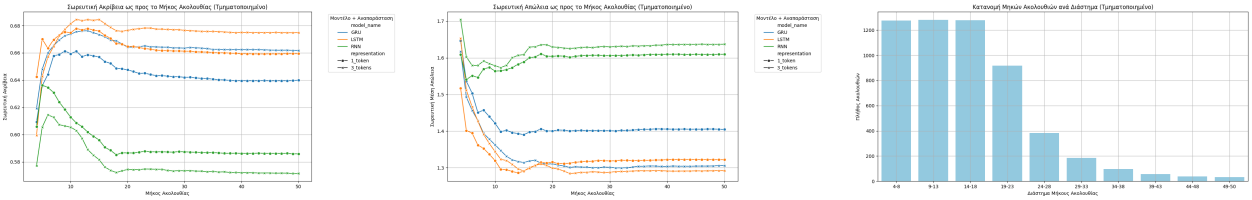
Figure 1.4.2: Σωρευτική Ακρίβεια και Απώλεια για το Τμηματοποιημένο σύνολο δεδομένων.

Table 1.3: Μέση ακρίβεια και απώλεια ανά μοντέλο και αναπαράσταση σε όλα τα σύνολα δεδομένων.

		Segmented		Labeled		Filtered	
Μοντέλο	**Αναπαράσταση**	**Μέση Ακ.**	**Μέση Απ.**	**Μέση Ακ.**	**Μέση Απ.**	**Μέση Ακ.**	**Μέση Απ.**
RNN	**1-token**	0.5859	1.63698	0.5632	1.6081	0.5448	1.6655
	3-tokens	0.5715	1.6370	0.5481	1.6504	0.5281	1.6972
LSTM	**1-token**	0.6594	1.3221	0.6244	1.3319	0.6007	1.4037
	3-tokens	**0.6748**	**1.2919**	**0.6394**	1.3079*	**0.6121**	1.3818*
GRU	**1-token**	0.6399	1.4046	0.6017	1.3832	0.5791	1.4479
	3-tokens	0.6618*	1.3060*	0.6311*	**1.2961**	0.6055*	**1.3782**
Mamba	**1-token**	0.5995	1.5165	-	-	-	-
	3-tokens	0.6501	1.4054	-	-	-	-

Τα καλύτερα αποτελέσματα είναι σε **έντονη γραφή**, τα δεύτερα καλύτερα είναι σημειωμένα με *.

Το σχήμα 1.4.3 απεικονίζει τον τρόπο με τον οποίο το μήκος της ακολουθίας εισόδου επηρεάζει την απόδοση της πρόβλεψης χρησιμοποιώντας τη μέση σωρευτική ακρίβεια και απώλεια. Η μέση αθροιστική ακρίβεια και απώλεια αντικατοπτρίζουν τον τρόπο με τον οποίο το μοντέλο επωφελείται από τη διαθεσιμότητα περισσότερων συγχορδιών πριν από την πραγματοποίηση μιας πρόβλεψης. Η αξιολόγηση σε αυτό το πείραμα διεξήχθη στο τμηματοποιημένο σύνολο δεδομένων. Πολύ σύντομες ακολουθίες, όπως αυτές μήκους 4, παρέχουν ανεπαρκή συμφραζόμενα, οδηγώντας σε χαμηλότερη ακρίβεια, ενώ οι μέτρια μακρύτερες ακολουθίες παρέχουν στο μοντέλο πλουσιότερα συμφραζόμενα και σημαντικά καλύτερες προβλέψεις. Τα απλά αναδρομικά νευρωνικά δίκτυα τείνουν να υποβαθμίζονται πέραν των ακολουθιών μήκους περίπου 7 συγχορδιών, ενώ άλλα μοντέλα συνεχίζουν να βελτιώνονται και να σταθεροποιούνται περίπου στις 15 συγχορδίες. Μετά από αυτό το σημείο, η προσθήκη περισσότερων συγχορδιών έχει μικρή επίδραση στην απόδοση.

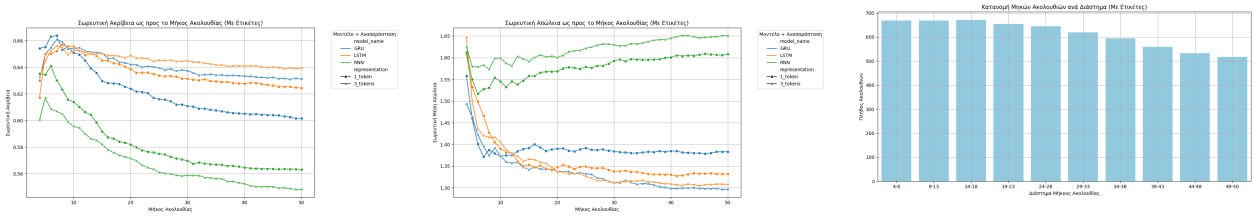


Σωρευτική Ακρίβεια (Τμηματοποιημένο) | Σωρευτική Απώλεια (Τμηματοποιημένο) | Κατανομή Μήκους Ακολουθιών (Τμηματοποιημένο)

Figure 1.4.3: Σωρευτική Ακρίβεια, Σωρευτική Απώλεια και Κατανομή Μήκους Ακολουθιών για το Τμηματοποιημένο σύνολο δεδομένων.

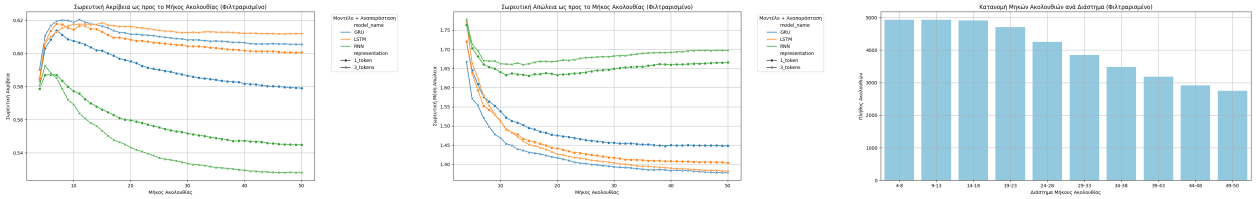
Είναι επίσης προφανές ότι η αναπαράσταση 3-όρων υπερτερεί της αναπαράστασης 1-όρου τόσο στα δίκτυα μακράς και βραχείας μνήμης όσο και στις αναδρομικές μονάδες με πύλες, ενώ στα απλά αναδρομικά νευρωνικά δίκτυα οι επιδόσεις είναι αντίστροφες. Το τελευταίο διάγραμμα δείχνει την κατανομή του μήκους των ακολουθιών ελέγχου, η οποία ευθυγραμμίζεται στενά με την κατανομή μεγέθους των τμημάτων που φαίνεται στο Σχήμα 1.3.5.

Στα Σχήματα 1.4.4 και 1.4.5, η αναπαράσταση 3-όρων υπερτερεί και πάλι της αναπαράστασης 1-όρου για τα δίκτυα μακράς και βραχείας μνήμης και τις επαναλαμβανόμενες μονάδες με πύλη, ενώ η αντίθετη τάση παρατηρείται για τα απλά αναδρομικά δίκτυα. Οι αξιολογήσεις που απεικονίζονται στα σχήματα διεξήχθησαν στο σύνολο δεδομένων με ετικέτες και στο φιλτραρισμένο σύνολο δεδομένων, αντίστοιχα. Σε σύγκριση με το τμηματοποιημένο σύνολο δεδομένων, οι κατανομές δειγμάτων σε αυτές τις αξιολογήσεις είναι πιο ισορροπημένες. Η ακρίβεια φαίνεται να μειώνεται καθώς τα μοντέλα λαμβάνουν μεγαλύτερες ακολουθίες συγχορδιών ως είσοδο, ενώ από την άλλη πλευρά η απώλεια γενικά μειώνεται με επιπλέον συγχορδίες, με τα αναδρομικά νευρωνικά δίκτυα να αποτελούν και πάλι την κύρια εξαίρεση. Συνολικά, τα μοντέλα φαίνεται να γίνονται πιο αξιόπιστα, αλλά όχι απαραίτητα πιο ακριβή.



Σωρευτική Ακρίβεια (Με Ετικέτες) Σωρευτική Απώλεια (Με Ετικέτες) Κατανομή Μήκους Ακολουθιών (Με Ετικέτες)

Figure 1.4.4: Σωρευτική Ακρίβεια, Σωρευτική Απώλεια και Κατανομή Μήκους Ακολουθιών για το σύνολο δεδομένων Με Ετικέτες.



Σωρευτική Ακρίβεια (Φιλτραρισμένο) Σωρευτική Απώλεια (Φιλτραρισμένο) Κατανομή Μήκους Ακολουθιών (Φιλτραρισμένο)

Figure 1.4.5: Σωρευτική Ακρίβεια, Σωρευτική Απώλεια και Κατανομή Μήκους Ακολουθιών για το Φιλτραρισμένο σύνολο δεδομένων.

Για καλύτερη κατανόηση της απόδοσης του μοντέλου, καταγράψαμε τα πιο συνηθισμένα, τα πιο επιζήμια και τα πιο σημαντικά λάθη σε όλα τα μοντέλα κατά τη διάρκεια της αξιολόγησης.

Ο Πίνακας 1.4 παραθέτει τα πιο συνηθισμένα λάθη πρόβλεψης. Πολλά ζεύγη συγχορδιών εμφανίζονται και στις δύο κατευθύνσεις (A→B και B→A). Η «μέση πρόσθετη απώλεια» αναφέρεται στη μέση αύξηση της απώλειας όταν μια συγχορδία προβλέπεται λανθασμένα, δηλαδή πόσο μικρότερη θα ήταν η συνολική απώλεια εάν είχε προβλεφθεί η πραγματική συγχορδία για αυτή την περίπτωση. Οι μέσες πρόσθετες απώλειες κυμαίνονται από 1,22 έως 1,88.

Στον πίνακα 1.5 παρουσιάζουμε τα πιο επιζήμια λάθη, που αντιπροσωπεύουν σπάνια σφάλματα τα οποία προκαλούν μεγάλες απώλειες όταν συμβαίνουν. Είναι ενδιαφέρον να σημειωθεί ότι αυτές οι συγχορδίες ανήκουν στο περιθώριο, καθώς προβλέπονται λανθασμένα μία φορά σε ολόκληρη την αξιολόγηση, ενώ δημιουργούν τεράστιο αντίκτυπο στην απώλεια. Η μέση προστιθέμενη απώλεια κυμαίνεται από 35,35 έως 65,10.

Τέλος, ο πίνακας 1.6 παρουσιάζει τα πιο σημαντικά λάθη συνολικά, συνδυάζοντας τη συχνότητα και τη σο-

βαρότητα. Οι περισσότερες από τις καταχωρήσεις εμφανίζονται επίσης μεταξύ των πιο συνηθισμένων λαθών, ενώ η μέση πρόσθετη απώλεια κυμαίνεται από 1,23 έως 1,93.

Table 1.4: Τα 20 πιο συνηθισμένα λάθη στις συγχορδίες

Πραγματική	Προβλεπόμενη	Συνολικός αριθμός	Μέση πρόσθετη απώλεια
C	G	13889	1.31
G	D	11526	1.23
G	C	11393	1.29
D	G	11369	1.31
D	A	9320	1.26
Emin	G	9032	1.81
F	C	8495	1.47
Amin	C	8467	1.63
C	D	7513	1.32
F	G	7144	1.46
A	D	7050	1.28
Amin	G	6905	1.88
D	C	6755	1.34
E	A	6729	1.37
C	F	6725	1.28
G	F	6584	1.22
G	A	6559	1.43
C	Amin	6128	1.34
Emin	C	5830	1.80
A	E	5785	1.19

1.5 Συμπεράσματα

1.5.1 Συζήτηση

Αυτό το κεφάλαιο ερμηνεύει τα αποτελέσματα των περιγραφικών και πειραματικών αναλύσεων σε σχέση με τους στόχους της μελέτης. Χρησιμοποιώντας το σύνολο δεδομένων Chordonomicon, η μελέτη εξετάζει εάν τα ερμηνεύσιμα, ελαφριά μοντέλα μπορούν να καταγράψουν σημαντικά αρμονικά και δομικά μοτίβα στη συμβολική μουσική. Συνδυάζοντας τη μεγάλης κλίμακας στατιστική εξερεύνηση με ακολουθιακή μοντελοποίηση, αποκαλύπτει τόσο τα κοινά τονικά θεμέλια όσο και τις εξαρτώμενες από τα συμφραζόμενα παραλλαγές μεταξύ των ειδών, των δεκαετιών και των τμημάτων των τραγουδιών. Η συζήτηση χωρίζεται σε δύο μέρη, την ανάλυση των δοκιμών και αρμονικών ιδιοτήτων του συνόλου δεδομένων και την ερμηνεία των πειραματικών αποτελεσμάτων.

Περιγραφική Ανάλυση

Το Chordonomicon δείχνει ότι η δημοφιλής μουσική βασίζεται σε ισχυρές αρμονικές κανονικότητες με λεπτές στυλιστικές παραλλαγές. Τα περισσότερα τραγούδια περιέχουν 6-8 τμήματα, με το μέσο μήκος τμήματος να είναι 8 συγχορδίες, ενώ τοπικά μέγιστα εμφανίζονται σε πολλαπλάσια του τέσσερα, αντανακλώντας τη συμβατική μετρική φρασεολογία. Οι στροφές και τα ρεφρέν κυριαρχούν τόσο σε συχνότητα όσο και σε μήκος, ενώ οι εισαγωγές και οι καταλήξεις λειτουργούν κυρίως ως σύντομες μεταβάσεις. Η Μέταλ και η Ραπ τείνουν να έχουν μακρύτερα τμήματα, με την Ραπ να αντισταθμίζει το γεγονός ότι έχει λιγότερα σε πλήθος τμήματα ανά τραγούδι κατά μέσο όρο, δείχνοντας προτιμήσεις σύνθεσης, συγκεκριμένες για κάθε είδος, μέσα σε ένα κοινό πλαίσιο. Συνολικά, το μήκος και η δομή των τμημάτων παραμένουν σταθερά σε όλα τα είδη και τις δεκαετίες, υποδηλώνοντας μια πολύ τυποποιημένη υποκείμενη δομή των τραγουδιών.

'Οσον αφορά το αρμονικό λεξιλόγιο, οι μείζονες και ελάσσονες τριάδες κυριαρχούν στο σύνολο δεδομένων του Chordonomicon (>90%), ενώ πιο σύνθετες ή ακανόνιστες συγχορδίες εμφανίζονται κυρίως στη Μέταλ, στη Ροκ και στην Πανκ. Η Τζαζ και η Σόουλ προτιμούν συγχορδίες με επεκτάσεις και περιστασιακές αναστροφές, αντανακλώντας μεγαλύτερη αρμονική εκφραστικότητα παρά τη σπανιότητά τους. Με την πάροδο του χρόνου, τα

Table 1.5: Τα 20 πιο οδυνηρά λάθη στις συγχορδίες ανά περίπτωση

Πραγματική	Προβλεπόμενη	Συνολικός αριθμός	Μέση πρόσθετη απώλεια
Fs/C	C7	1	65.10
Eb/A	Fsmin	1	56.38
Adim/C	Gmin	1	49.49
F/B	Dmin	1	48.71
Gs/B	G	1	42.50
Cmin7/A	Eb	1	41.86
Gb11	G	1	41.79
Gsus4/G	Fmaj7	1	40.59
Csmin/D	Csmin	2	39.73
Cminmaj7	Asno3d	1	39.05
Fsus4/E	Fs	1	38.95
Gbminadd13	Dmin	1	38.08
Aminadd13/E	E	1	37.04
Fsmin11/A	A	1	36.99
Bbsus4/G	Eb	1	36.82
Bb/As	Emin	1	36.57
Fmaj9/C	Cmaj7/G	1	36.35
Cdim/Eb	F7	1	36.29
Gsmin13	Ddim7	1	36.25
Emin9/G	Badd9/Ds	1	35.34

δεδομένα αποκαλύπτουν μια σταδιακή απλοποίηση της αρμονίας, με μειωμένη ποικιλία συγχορδιών τις τελευταίες δεκαετίες, ανταναικλώντας, πιθανώς, αλλαγές στις τεχνικές παραγωγής, την αισθητική και τις δημοφιλείς τάσεις.

Εξετάζοντας τα ακολουθιακά χαρακτηριστικά, οι κινήσεις των ριζών στο Chordonomicon ακολουθούν στενά τον κύκλο των πέμπτων, οι κινήσεις των 5 ημιτονίων είναι οι πιο συχνές, ακολουθούμενες από βήματα 2 ημιτονίων, ενώ οι μεταβάσεις 6 ημιτονίων είναι σπάνιες. Αυτά τα μοτίβα ευθυγραμμίζονται με κοινές τονικές προόδους όπως I-IV-V-I και δεν παρουσιάζουν κατευθυντική μεροληψία μεταξύ ανοδικών και καθοδικών μεταβάσεων. Η συνέπεια αυτών των τάσεων σε όλα τα είδη και τις δεκαετίες υπογραμμίζει την τονική σταθερότητα ως βασικό χαρακτηριστικό της δημοφιλούς μουσικής.

Επιπροσθέτως, οι αναλύσεις ακολουθιών n-στοιχείων δείχνουν ότι οι μεταβάσεις των συγχορδιών ποικίλλουν στα όρια των τμημάτων, με τα μοτίβα δυάδων και τριάδων να διαφέρουν από αυτά των συνολικών μεταβάσεων, Η μέτρηση της συνολικής απόστασης διακύμανσης επιβεβαιώνει ότι οι αρμονικοί κανόνες μεταβάλλονται συστηματικά κατά τη μετάβαση μεταξύ των δομικών τμημάτων. Αυτό υποδηλώνει ότι οι τοπικές ακολουθίες συγχορδιών περιορίζονται από τα όρια των μερών, δηλαδή ότι τα μεγέθη των τμημάτων περιέχουν πλήρεις αρμονικές φράσεις, τονίζοντας έτσι τη σημασία της ενσωμάτωσης του δομικού πλαισίου στην προγνωστική μοντελοποίηση των συμβολικών ακολουθιών συγχορδιών.

Επεκτείνοντας τις αναλύσεις σε ακολουθίες μήκους 4, 5 και 6 συγχορδιών, χρησιμοποιήθηκαν μοντέλα λανθάνουσας απόστασης για την αναπαράσταση των τραγουδιών μέσω γραφημάτων, παρέχοντας μια ερμηνεύσιμη οπτικοποίηση των μοτίβων που είναι συγκεκριμένα για κάθε είδος και δεκαετία, μέσω ομαδοποίησης. Η ενσωμάτωση των πλειάδων σε έναν κοινό λανθάνοντα χώρο αποκαλύπτει στατιστική εγγύτητα μεταξύ των ειδών και των χρονικά γειτονικών δεκαετιών, επιβεβαιώνοντας ότι τα στατιστικά των πλειάδων καταγράφουν ορισμένες σχέσεις μεταξύ των χρονικών μοτίβων. Ακόμα, οι περισσότερες πλειάδες καταλαμβάνουν πυκνές κεντρικές περιοχές, ανταναικλώντας κοινά αρμονικά μοτίβα, ενώ οι στυλιστικές και χρονικές παραλλαγές αναδύονται διακριτικά σε τοπικές δομές υψηλότερης τάξης, υποστηρίζοντας την καθολικότητα των αρμονικών θεμελίων.

Πειράματα

Συμπληρώνοντας αυτές τις παρατηρήσεις, τα πειράματα ακολουθιακής μοντελοποίησης, σχετικά με την ταξινόμηση ανά είδος και δεκαετία και την πρόβλεψη επόμενων συγχορδιών, παρουσιάζουν ενδιαφέροντα αποτελέσματα.

Table 1.6: Τα 20 πιο σημαντικά λάθη στις συγχορδίες συνολικά

Πραγματική	Προβλεπόμενη	Συνολικός αριθμός	Μέση πρόσθετη απώλεια	Συνολική πρόσθετη απώλεια
C	G	13889	1.31	17854
Emin	G	9032	1.81	16129
D	G	11369	1.31	14536
G	C	11393	1.29	14409
G	D	11526	1.23	13964
Amin	C	8467	1.63	13744
Amin	G	6905	1.88	13053
F	C	8495	1.47	12179
D	A	9320	1.26	11466
Emin	C	5830	1.80	10193
F	G	7144	1.46	10078
C	D	7513	1.32	10002
Bmin	D	5537	1.80	9982
Emin	D	5171	1.93	9973
D	C	6755	1.34	9157
E	A	6729	1.37	9089
G	A	6559	1.43	8949
A	D	7050	1.28	8929
C	F	6725	1.28	8597
C	Amin	6128	1.34	8258

Για τα προβλήματα ταξινόμησης, τα ελαφριά ακολουθιακά μοντέλα, τα μοντέλα κατάστασης-χώρου και τα μοντέλα λανθάνουσας απόστασης είχαν χαμηλή απόδοση. Μοντέλα τα οποία εκπαιδεύτηκαν χρησιμοποιώντας αναπαραστάσεις βασισμένες μόνο σε τονικές κλάσεις πέτυχαν μέτρια αποτελέσματα, ενώ τα απλά αναδρομικά νευρωνικά δίκτυα και τα μοντέλα λανθάνουσας απόστασης είχαν αποτελέσματα που προσομοίαζαν την τυχαία επιλογή. Οι πιο προηγμένες αρχιτεκτονικές, όπως οι παραλλαγές των απλών αναδρομικών νευρωνικών που χρησιμοποιούν μηχανισμούς με πύλες και Mamba, είχαν καλύτερη απόδοση, αλλά εξακολουθούσαν να αντιμετωπίζουν δυσκολίες, αντανακλώντας την περιορισμένη διακριτική ικανότητα των αρμονικών χαρακτηριστικών στα διάφορα είδη και δεκαετίες.

Παρά την ασθενή απόδοση της ταξινόμησης, τα πειράματα παρέχουν πολύτιμες πληροφορίες. Οι πίνακες σύγχυσης, από την ακολουθιακή μοντελοποίηση, δείχνουν ότι οι λανθασμένες ταξινομήσεις συμβαίνουν συχνά μεταξύ κατηγοριών με παρόμοιο στυλ, σε συμφωνία με τις εγγύτητες στον λανθάνοντα χώρο των ακολουθιών μήκους n. Αυτό ενισχύει την ιδέα ότι μια κοινή αρμονική δομή εκτείνεται σε διάφορα είδη και δεκαετίες, αντανακλώντας σταθερά θεμέλια με λεπτές παραλλαγές.

Τα πειράματα πρόβλεψης συγχορδιών υπογραμμίζουν τη σημασία των αρμονικών συμφραζόμενων και της δομής σε επίπεδο τμημάτων. Σε όλες τις αρχιτεκτονικές η απόδοση βελτιώνεται ραγδαία με την αύξηση των δεδομένων εκπαίδευσης έως περίπου 200.000 δείγματα διαδοχών συγχορδιών, μετά τα οποία η βελτίωση σταθεροποιείται. Αυτό υποδηλώνει ένα σημείο κορεσμού όπου τα μοντέλα έχουν καταγράψει τις κυρίαρχες αρμονικές εξαρτήσεις και τα περεταίρω πρόσθετα δεδομένα προσφέρουν μόνο οριακές βελτιώσεις.

Η ανάλυση της απόδοσης ως προς το μήκος της ακολουθίας πριν την πρόβλεψη δείχνει ότι η απόδοση βελτιώνεται με το μεγαλύτερο πλήθος συμφραζόμενων έως περίπου 15 συγχορδίες, μετά τις οποίες τα οφέλη σταθεροποιούνται. Αυτό υποδηλώνει ότι οι περισσότερες προγνωστικές πληροφορίες είναι τοπικές. Η αναπαράσταση συγχορδιών με 3 όρους (ρίζα, ποιότητα + επεκτάσεις, αναστροφή), υπερτερεί σταθερά της προσέγγισης με 1 όρο, καθώς βοηθά τα μοντέλα να μάθουν τις σχέσεις μεταξύ των συστατικών των συγχορδιών αντί να απομνημονεύουν ολόκληρα σύμβολα, αποδεικνύοντας ότι η αρμονική κατανόηση ωφελείται από τις σύνθετες αναπαραστάσεις συγχορδιών.

Τα απλά αναδρομικά νευρωνικά δίκτυα έχουν χειρότερη απόδοση, από τις παραλλαγές τους που χρησιμοποιούν πύλες, λόγω της περιορισμένης ικανότητάς τους να διατηρούν μακροπρόθεσμα τα σχετικά συμφραζόμενα ώστε να μοντελοποιήσουν σύνθετες αρμονικές εξαρτήσεις. Είναι επίσης ενδιαφέρον ότι αντιστρέφεται η διαφορά της απόδοσης μεταξύ της προσέγγισης με 3 όρους και αυτής με 1 όρο, καθώς η ανεξάρτητη πρόβλεψη κάθε

συστατικού της συγχορδίας συνεισφέρει στην συσσώρευση αβεβαιότητας όταν τα συνδυάζονται για την τελική πρόβλεψη, αυξάνοντας το συνολικό σφάλμα.

Τέλος, όταν τα μοντέλα δοκιμάζονται σε ολόκληρα τραγούδια, αγνοώντας την εσωτερική τμηματοποίησή τους, η απόδοση μειώνεται. Οι μακρύτερες ακολουθίες εισόδου βελτιώνουν ελαφρώς την απώλεια, αλλά η ακρίβεια μειώνεται καθώς μετά από μεταβάσεις σε νέο τμήμα τα συμφραζόμενα του προηγουμένου χάνουν αρκετά την αξία τους, προκαλώντας ασυμβίβαστες ακολουθίες και συστηματικά σφάλματα. Μειώνεται η ικανότητα πρόβλεψης των μοντέλων ωστόσο τα παλαιότερα συμφραζόμενα ενισχύουν ελαφρώς την αξιοπιστία τους. Αυτό υπογραμμίζει τη σημασία της δομής σε επίπεδο των τμημάτων στη μοντελοποίηση συμβολικών διαδοχών συγχορδιών. Η απόδοση ποικίλλει επίσης μεταξύ συνόλων δεδομένων, με τα επιμελημένα κείμενα που περιέχουν πληροφορίες τμηματοποίησης να αποδίδουν πιο αξιόπιστα αποτελέσματα.

1.5.2 Περιορισμοί και Μελλοντική Έρευνα

Αν και η μελέτη αυτή παρέχει πληροφορίες για τις συμβολικές διαδοχές συγχορδιών, υπάρχουν αρκετοί περιορισμοί που υπογραμμίζουν τις προκλήσεις και τις ευκαιρίες για μελλοντική έρευνα.

- **Αλλαγές στην αξία των συμφραζόμενων:** Η ικανότητα πρόβλεψης της επόμενης συγχορδίας μειώνεται στα όρια των τμημάτων. Μελλοντικές μελέτες θα μπορούσαν να διερευνήσουν την Ανίχνευση Τμημάτων χρησιμοποιώντας την αβεβαιότητα του μοντέλου που προκύπτει μετά τις αλλαγές, ενώ και το πρόβλημα της πρόβλεψης επόμενης συγχορδίας μπορεί να επωφεληθεί γνωρίζοντας πότε προκύπτουν μεταβάσεις.
- **Μη ισορροπημένο σύνολο δεδομένων και η επιμέλεια του:** Η εξισορρόπηση του Chordonomicon για ταξινόμηση απέκλεισε πολλές καταχωρήσεις, με πιθανή απώλεια μοτίβων. Προσεγγίσεις όπως χρήση σταθμισμένων απωλειών και η τεχνητή αύξηση των δεδομένων θα μπορούσαν να διατηρήσουν την αρμονική και στυλιστική ποικιλομορφία.
- **Περιορισμοί αποκλειστικής χρήσης συμβολικών συγχορδιών:** Οι συμβολικές διαδοχές συγχορδιών παραλείπουν στοιχεία όπως τη μελωδία, τον ρυθμό, το τέμπο αλλά και την διάρκειά τους. Πολυτροπικές προσεγγίσεις, ή πρόσθετες πληροφορίες, θα μπορούσαν να βελτιώσουν την απόδοση τόσο στο πρόβλημα της ταξινόμησης όσο και στο πρόβλημα της πρόβλεψης.
- **Λανθασμένες προβλέψεις συγχορδιών:** Τα σφάλματα των μοντέλων περιλαμβάνουν είτε σπάνιες συγχορδίες με υψηλές απώλειες ή συχνές και μουσικά εύλογες συγχύσεις (όπως Cmaj έναντι Gmaj ή Cmaj έναντι Amin). Τεχνικές εξομάλυνσης ή χρήση ειδικά προσαρμοσμένων απωλειών θα μπορούσαν να λάβουν υπόψη μουσικά παρόμοιες εναλλακτικές λύσεις, μειώνοντας τις ποινές για αποδεκτές προβλέψεις, δηλαδή και την επίδρασή τους κατά τη διάρκεια της εκπαίδευσης.

Chapter 2

Introduction

Music is one of the most expressive forms of human creativity, its structure and internal rules have captured the attentions of composers, scientists and technologists alike, for centuries. Understanding how harmony, melody, and rhythm interact to create emotion, express ideas and tell stories, has long been an important aim in the domains of art and science. With the rapid expansion of digital music data and the evolution of computational tools, the study of musical structure has increasingly transitioned from pure theoretical analysis to data-driven approaches. The increasing interest and the improvement of modeling mechanisms have given way to the field of Musical Information Retrieval (MIR), that aims to analyze, model, and understand music, using computational and statistical methods.

At the core of musical sequences lie the chords and their progressions. They form the most fundamental elements for studying tonal and stylistic characteristics. These progressions define harmonic motion, establishing the patterns of tension and resolution that organize musical form and contribute to its character. By modeling them, it becomes possible to track the historical evolution of harmonic language, uncover stylistic signatures across different genres and learn to predict musical events such as what chords or modulations may follow.

However, while chord progressions capture essential harmonic information, they represent only one dimension of the musical structure. Symbolic chord progressions depict purely harmonic abstractions, they do not completely encode melodic movement, rhythmic structure, tempo or expressive timing. Focusing totally on music's tonal backbone makes symbolic chord progressions a concise yet constrained way to describe music including rich harmonical representation.

Traditional approaches to harmonic analysis have relied more on symbolic rule-based methods or shallow statistical models that capture only local relationships between chords. The recent emergence of machine learning and deep learning methods has created a new path towards understanding music. Neural architectures suitable for sequential modeling such as, recurrent neural networks with their gated variants, Transformer-based models and more recently, state-space models like Mamba, have introduced the capacity needed to capture long-range dependencies and hierarchical structures in musical sequences. These methods enable learning harmonic and stylistic relationships, extracted from raw data, without the need for handcrafted rules. The goal of this study is to model and analyze musical structure through symbolic chord progressions using both statistical and machine learning methods, bridging the gap between music theory and computational modeling and encoding the harmonic and temporal properties of music.

More specifically this thesis aims to:

- Investigate how symbolic representations of chords and progressions can effectively describe musical structure across genres and decades.
- Employ and compare statistical models and deep learning architectures for modeling chord sequences
- Evaluate model inference on key music information retrieval tasks such as genre and decade classification and chord prediction

- Analyze how model performance reflects harmonic similarities, stylistic grouping and temporal evolution

By integrating the theoretical foundations of music with modern machine learning techniques, this work contributes to a deeper understanding of how data-driven models can capture the essence of harmonic organization. Furthermore, it provides insights into how computational representations of music align with human perception and theory, offering perspective for future applications in automatic composition, music recommendation and stylistic analysis.

The remainder of this thesis is organized as follows. It begins by introducing the foundational principles of music theory and symbolic representations relevant to computational analysis. Next, it provides an overview of machine learning principles, followed by a detailed description of sequential models that utilize recurrent and attention-based mechanisms. The work then presents the Latent Distance Model, further explaining the proposed research framework. Finally, the following sections describe the dataset, feature extraction process, and preliminary analysis, before detailing the experimental setup, model evaluation, and results. The thesis concludes with a discussion of the finding, their limitations, and potential directions for future research.

Chapter 3

Music Theory

Music is a form of organized sound that uses different elements to express emotions, spread ideas and tell stories. The desire to understand the structure and relationships behind it eventually gave rise to music theory. Music theory provides the formal language and rules that create an analytical framework for understanding music. Concepts such as pitch, melody, rhythm, harmony, and timbre are crucial to how music is composed, performed, and perceived. In the context of Music Information Retrieval (MIR), these concepts are not only informative but also usable by computational systems. Music Information Retrieval is the field that studies music through computer science and data analysis. Its goal is to analyze, understand and organize music. Besides using raw audio for such analysis, representing musical elements symbolically, through notes, chords, progressions, or rhythms, enables to algorithmically analyze, model and understand music across a wide range of tasks.

Every music information retrieval task benefits from a strong foundational knowledge of music theory. Tasks like melody extraction, chord recognition, key detection, genre classification, cover song detection, or similarity measurement depend on capturing musical structure and relationships. Understanding harmonic function, melodic contour, tonal hierarchy, and tonal patterns opens the door for computational models to learn important patterns that reflect human musical perception and cognition.

By integrating music theory into complex computational models, music information retrieval systems gain a stronger analytical foundation, moving beyond purely statistical methods, and capturing musically meaningful patterns. Symbolic representations of musical elements provide a way for machine learning models to absorb theoretical knowledge and analyze music using its essential structural components. This chapter provides the necessary theoretical background [1] to understand the role of music theory and its influence on the design, interpretation, and evaluation of music information retrieval problems across various tasks.

Contents

3.1 Pitch and Intervals **42**
3.2 Scales and Keys **42**
3.3 Chord, harmony and Chord Progressions **43**
3.4 Symbolic Representations for Computational Modeling **44**
3.5 Applications to Music Information Retrieval (MIR) Tasks **44**
3.5.1 Genre and Decade Classification 44
3.5.2 Chord Prediction 45
3.5.3 Other tasks 45

3.1 Pitch and Intervals

At a foundational level, music is organized around frequency. Frequency determines the highness or lowness of a perceived sound. In music the quality of a sound governed by the vibrations producing it is called pitch. In Western tonal music, pitches, whose ratio in frequency is $2:1$, are considered octave equivalents, an octave being the interval between these two pitches. This perceptual property makes pitches groupable into pitch classes, allowing the containment of the theoretically infinite frequency spectrum into twelve discrete categories within an octave. These pitch classes, also called notes in an octave, form the basis of most symbolic music representations used in music information retrieval tasks. The twelve pitch classes are $C, C\#/D\flat, D, D\#/E\flat, E, F, F\#/G\flat, G, G\#/Ab, A, A\#/B\flat$ and B.

An interval is the distance between two pitches and in the Western system this distance is measured in semitones, the smallest equal division of the octave. For instance, the distance between C and E consists of four semitones and is labeled a major third, while the interval between C and G consists of seven semitones and is called a perfect fifth. Intervals are used to describe both quantitative and qualitative relationships between pitches.

Understanding pitch and interval structure is an important step to move from absolute frequency to relative pitch information. Features such as chroma vectors or pitch-class profiles do not rely on octave information, they categorize pitches according to pitch classes providing transposition invariance and focusing on patterns based on chord progressions or melodic contours.

In symbolic music information retrieval, pitches and intervals are commonly encoded using integers. An example is the C major triad being represented by the pitch-class set $\{0, 4, 7\}$, corresponding to the root, the C note, the major third, the E note, and the perfect fifth, the G note. On the other hand, instead of pitch classes, interval representation can also be used, such as $\{+4, +3\}$ for the same triad, denoting the way the notes move in it. Representations of this category prioritize structure over content, allowing machine learning models to learn patterns based on motion, spacing and harmonic relations leaving tonal contexts aside.

This abstraction of musical data to pitch and interval representations helps music information retrieval systems to align with theoretical descriptions of melody and harmony. It also serves as a first step to understanding and modeling more complex concepts, like scales, keys, chords, and progressions.

3.2 Scales and Keys

In tonal music, where a central note acts as a base for the system, melody and harmony are constructed on sequences of pitches. Organized pitch sequences are called scales. In Western music, the most common scales are the major and minor and they are defined by specific patterns of whole and half steps, tones and semitones, between successive notes. The major scale follows the pattern W–W–H–W–W–W–H, with W denoting the whole step and H the half step. For example, the C major scale based around the C pitch is composed of C–D–E–F–G–A–B. The minor scale's pattern is W–H–W–W–H–W–W including scales like the A minor (Aeolian) scale that comprises the pitch classes A–B–C–D–E–F–G. These two scales consist of the same pitch classes but have a different key.

A key refers to the tonal center of a musical piece, normally the tonic note of a scale, which serves as a point of resolution and stability. The key creates a hierarchy amongst pitches, assigning roles to notes and chords, some being more stable or tense relative to the tonic. These roles like tonic, dominant, and subdominant describe the function of each part of the scale and are important for understanding harmonic progression. Tonic is the main note of the scale, the fifth note is the dominant and the fourth the subdominant with the subdominant being the same distance below the tonic as the dominant is above it. The rest of the notes also have descriptive names, the third is the mediant, the sixth the submediant the second is the supertonic and the seventh is the called either leading tone or subtonic depending on the scale [16].

A systematic way to visualize relationships between keys is the circle of fifths. The circle of fifths shows how each key is related to the others by perfect fifth intervals of their tonics. Viewed in a counterclockwise direction it represents a circle of fourths. Keys closer on the circle share many common pitches, which explains why modulations, being shifts of key, between neighboring keys sound more natural. Understanding

key relationships is important for music information retrieval tasks such as key detection, harmonic similarity, and transposition-invariant pattern detection, where the same patterns can appear in different keys.

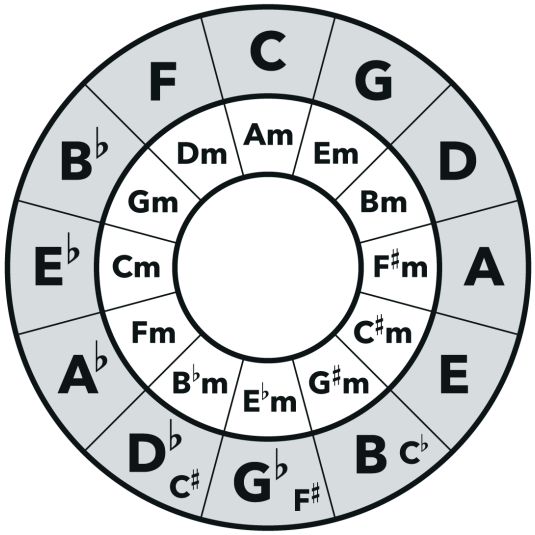


Figure 3.2.1: Circle of Fifths, important for understanding relationships between key signatures [2].

For computational analysis, scales and keys are often represented symbolically, as pitch class sets or as sequences of scale degrees with respect to the tonic. These representations absorb scale and key information allowing for another layer of abstraction helping music information retrieval systems analyze melodies and harmonic structures in a key-independent manner, while capturing the tonal substructure of musical pieces.

3.3 Chord, harmony and Chord Progressions

Chords are sets of pitches sounded together. The simplest form of chords is that of triads. Each triad consists of the root, the third and the fifth, while there exist variants that replace the third or completely remove it. The C major triad, for example, includes the notes C (root), E (major third), G (perfect fifth). Chords can be extended by adding additional notes such as sevenths, ninths, or other extensions that enrich their harmonic color.

The structure of each chord is classified based on its quality. The quality defines the intervals between the root and the other tones in the chord. The most common qualities are shown below:

Major Triad: root, major third, perfect fifth (C major: *C–E–G*)

Minor Triad: root, minor third, perfect fifth (A minor: *A–C–E*)

Augmented Triad: root, major third, augmented fifth (C augmented: *C–E–G♯*)

Diminished Triad: root, minor third, diminished fifth (A diminished: *A–C–E♭*)

Additionally, a chord can also appear in different inversions, keeping its notes but rearranging them so the lowest pitch in the chord, the bass, is other than the root. For instance, taking the first inversion of the C major triad, we end with E in the bass *E–G–C*, while the second inversion brings the G in the bass *G–C–E*. During inversion, the pitch content stays the same, what changes is the bass which can lead to smoother harmonic motion when progressing through chords.

In a musical piece, chords do not just stand by themselves, they interact with other chords over time. Harmony gives insights into their function and meaning in a sequence and provides the structural shape in which they express tension and resolution, setting the direction and adding emotional content in music. These relationships are what create a sense of motion, stability, or anticipation as time evolves.

Smooth chord transitions are closely related to the concept of voice leading. Voice leading refers to the stepwise movement of individual notes between successive chords which accounts for the perception of smooth harmonic transitions. For example, moving between chords that share some notes, like from C major (C–E–G) to A minor (A–C–E) results in a smoother transition than changing them all at once, like from C major to D minor (D–F–A).

Besides local chord relationships, multiple chords can be combined and form a progression, that are longer sequences that define tonal structure to a bigger extent. Common patterns such as I–IV–V–I and ii–V–I create cycles of tension and resolution, while less conventional sequences add stylistic variety. Functional harmony establishes roles to chords based on the scale degree of the note they were built on. The tonic (I) provides stability and resolution, the dominant (V) creates tension and seeks to resolve back to tonic and the subdominant (IV) includes variety and often transitions to dominant. Other scale degree chords (ii, iii, vi, vii^o) introduce color and variety while also influence directional tendencies.

In music information retrieval, harmonic information can be encoded symbolically using chord labels, sets of pitch classes or representations of the intervals between chord. Models suited for sequential analysis, such as those based on recurrence (long short-term memory systems and mamba) or those that capture sequential using other methods (self-attention in Transformers), can utilize these encoding and extract important features for tasks including chord recognition, chord prediction, key detection, genre or decade classification and similarity based on harmonic structure. Music information retrieval systems can model both local and global relationships in tonal harmony, by integrating knowledge on chord composition, quality, inversions, voice leading and progressions.

3.4 Symbolic Representations for Computational Modeling

Symbolic music representations encode features like pitch, duration and harmony as categorical or numeric values, providing a different way for computational models to learn musical context besides raw audio data.

Different formats include Musical Instrument Digital Interface (MIDI) files which represent notes by pitch numbers, duration and onset while chord annotations describe harmonic content through root, quality, extensions and bass. These symbols are often turned into numeric encodings and tokenized for learning. Chords can be expressed as one-hot vectors, pitch-class sets or chroma features than can be derived from audio signals and be represented symbolically.

To capture temporal and harmonic relationships of chord progressions sequences can be modeled using n-grams or embedding spaces, having similar chords and progressions occupy neighboring positions. In addition, architectures suited for sequential modelling, such as recurrent neural networks, transformers and state space models can derive information about progressions and learn dependencies, predict future chords and characterize tonal behavior.

Symbolic representations allow computational models to incorporate music theory into their learning process and enable them to analyze the structural properties of tonal harmony, moving beyond static representations.

3.5 Applications to Music Information Retrieval (MIR) Tasks

The theoretical concepts and symbolic representations described in this chapter create a strong basis for multiple music information retrieval tasks. Presented below are the most relevant task with this thesis.

3.5.1 Genre and Decade Classification

The goal of classification is to assign the correct label or labels to elements from the data. Genre classification aims to assign predefined musical styles, known as genres, to musical pieces. Features useful for this task may include pitch content, chord distributions, or harmonic progressions detected in chord sequences. Symbolic representations of chords and their progressions help identify relationships between each part of the sequence as well as between sequences. Some examples are the use of extended chords in jazz or repetitions in pop music. Although there exist some patterns, each genre adopts different elements that might also be shared amongst them, making the task a difficult one.

Similarly, decade classification aims to determine the time period of a musical piece by relying on stylistic and harmonic elements. Chord usage, tendencies in specific progression patterns and certain harmonic preferences can be indicators of temporal trends in music, allowing models to locate their place in time. Decade classification's goal is to describe how music evolves through time and is part of tasks concerning musical style evolution and similarity analysis.

3.5.2 Chord Prediction

Chord Prediction involves estimating the next chord in a sequence by leveraging the harmonic context provided. Inputs are typically symbolic chord labels, and the output is the most probable chord to follow the given preceding sequence. It essentially is a sequence modeling problem suitable for recurrent or attention-based architectures. Chord prediction is a task closely related to the tasks of chord recognition, which aims to automatically identify the chord sequence mostly based on audio input, and acts as a complementary. Chord progression can also be viewed from another angle as a chord generation task, as both rely on understanding harmonic motion and modelling musical style. Chord generation has different evaluation methods as multiple generated progressions may be musically valid depending on stylistic user-defined criteria.

3.5.3 Other tasks

While escaping the scope of this thesis, symbolic representations of music also support tasks such as key detection, beat tracking, melody extraction, harmonic analysis, music similarity, style identification and more.

This chapter establishes the scope of symbolic music information retrieval tasks, addressed in this thesis, along with the necessary theory that accompanies them, providing context for the modeling approaches presented in Chapter 9.

Chapter 4

Machine Learning

The idea of creating an entity with human-like characteristics and intelligence, one that can think for itself, has existed since ancient times. In the 20th century, this vision took a new form with the invention of programmable computers, leading to the establishment of Artificial Intelligence as a formal field. Early Artificial Intelligence systems focused primarily on problems that could be described with strict mathematical rules, relying on comprehensive knowledge bases to solve them. Interestingly, such problems often seem difficult for humans but easy for machines thanks to their computational power. However, these systems struggled with tasks that humans find intuitive, such as recognizing speech or images. The difficulty of formalizing such tasks with explicit rules quickly revealed the limitations of traditional Artificial Intelligence approaches [17].

To overcome these challenges, Machine Learning, a branch of Artificial Intelligence, emerged. Machine Learning introduced algorithms capable of extracting patterns and knowledge from training data, then using that knowledge to make inferences about new, unseen data. Instead of requiring explicit programming for every possible situation, Machine Learning models learn from experience, identifying statistical relationships within data and generalizing from them [18].

The more intuitive a task is for humans, the harder it often is to break down into formal steps or explicit rules. Machine Learning tasks vary in difficulty and are strongly influenced by the quantity, quality, and format of the available input data. Common input types include text, audio, images, time-series, graphs, or even a combination of these (multimodal data). Core tasks in Machine Learning include classifying data into predefined categories, predicting continuous numerical values through regression, or finding and grouping similar data points through clustering. The choice of a suitable learning method depends on the nature of the data and the specific goal of the task.

As a foundation for understanding these methods, it is important to examine the main Machine Learning paradigms, which define how models learn from data and the type of feedback they receive during training.

Contents

4.1 Learning Paradigms 49
4.1.1 Supervised Learning 49
4.1.2 Unsupervised Learning 49
4.1.3 Reinforcement Learning 49
4.1.4 Semi-Supervised Learning 49
4.1.5 Self-Supervised Learning 49
4.2 Neural Network Fundamentals 50
4.2.1 Perceptrons and Multilayer Networks 50
4.2.2 Activation Functions 51
4.2.3 Loss, Optimization and Backpropagation 51
4.3 Training and Generalization 52
4.4 Deep Learning 53

4.5 Tokenization, Embeddings and Feature Spaces 54

4.1 Learning Paradigms

Machine Learning algorithms are categorized by the way they extract knowledge from the data and the type of feedback they receive during training based on their performance. These algorithms fall into five broad categories: supervised learning, unsupervised learning, reinforcement learning, semi-supervised learning, and self-supervised learning.

4.1.1 Supervised Learning

In Supervised Learning the model is trained to make correct predictions based on the given input. This method is applied when a correct output already exists, an external "ground truth" labelling the data, which allows the model to compare its predictions and better its accuracy over time. The most common tasks under this paradigm are those of classification and regression. Classification involves predicting categorical outcomes, where the target variable represents distinct groups or labels while Regression deals with continuous target variables. Such approaches are widely used in Music Information Retrieval (MIR) for example genre classification [14] and music popularity prediction [19].

4.1.2 Unsupervised Learning

Unsupervised Learning algorithms do not involve any external labels guiding training. Instead, they focus on detecting intrinsic structures dependencies and correlations in the raw data. With no explicit instructions, the model is left to detect hidden patterns on its own. Common Unsupervised learning tasks are clustering, dimensionality reduction, and association. Clustering involves grouping data into clusters based on similarities found in their features. Dimensionality reduction is the process of reducing the number of dimensions used to represent the data while preserving essential information about them. Association is a technique for identifying rules that indicate correlation between features. These methods are used in song similarity analysis and music recommendation systems [20].

4.1.3 Reinforcement Learning

In Reinforcement Learning, an adaptive algorithm, known as an agent, learns by exploring the environment trying to reach a predefined goal. The agent is encouraged to learn the best strategies to achieve this goal, receiving feedback through rewards or penalties depending on an established metric. Over repeated interactions, through trial and error, the agent can distinguish the best strategies [21]. Common applications of Reinforcement Learning include robotics, autonomous systems, and game playing, where the agent must make sequential decisions to maximize long-term rewards.

4.1.4 Semi-Supervised Learning

Semi-Supervised Learning stands between supervised and unsupervised learning. This type of training is used when the dataset being used consists of a small, labeled part and a large, unlabeled part. During training, the algorithm uses the labeled data, indicatively, to guide its understanding of the unlabeled data. One such model might use unsupervised learning to detect clusters in the data and after that use the labeled data within each cluster to infer the labels of others [21]. Semi-Supervised Learning is commonly applied in situations where labeling the data is time-consuming or costly, such as speech recognition and text classification.

4.1.5 Self-Supervised Learning

Self-Supervised Learning is related to Semi-Supervised Learning, but it does not rely on human provided labels during training. It can be viewed as a form of Supervised Learning on unlabeled data. Instead of using external labels, the algorithm generates its own pseudo labels after training on the unlabeled data, learning internal representations. In some cases, small amounts of labeled data might be later used to fine-tune the model for specific real-world applications [18]. Self-Supervised Learning is widely used in cases where large volumes of unlabeled data are available, particularly in representation learning tasks such as natural language processing, computer vision, and music feature extraction.

Only Supervised and Unsupervised Learning will be considered in this thesis.

4.2 Neural Network Fundamentals

Neural Networks (NNs) are an important part of Machine Learning and are inspired by the structure and function of biological neural networks, such as the human brain. Artificial Neural Networks (ANNs) consist of interconnected computational units, called neurons or nodes, which are organized in layers. Their goal is to process input data and learn a mapping to the desired outputs.

Each neuron receives input, combines it with weights, applies an activation function, and passes the result to the next layer. During training, the network adjusts its internal parameters, weights, and biases, allowing it to recognize complex patterns and approximate functions that can accurately predict the outputs.

Even though these concepts are mostly associated with supervised learning, they are important for this thesis because they provide the foundation for understanding how neural networks, including recurrent models, operate. The rest of this section breaks down the core components of a neural network: how individual neurons function, what activation functions are, how networks learn through loss and optimization and how information flows and updates through layers. The information presented below was primarily sourced from [17], [22], [23].

4.2.1 Perceptrons and Multilayer Networks

As mentioned above, at its simplest, a neural network is made up of neurons, also called perceptrons. Each perceptron takes an input vector of features $X = [x_1, x_2, \ldots, x_n]^T$, multiplies each component by a corresponding weight $W = [w_1, w_2, \ldots, w_n]^T$ to control the influence of each input feature on the output, adds a bias b to shift the activation threshold, and transforms the result using an activation function ϕ producing an output y:

$$y = \phi\Big(\sum_{i=1}^{n} w_i x_i + b\Big) = \phi(W^T X + b) \tag{4.2.1}$$

A perceptron is a simple classifier. The weights and bias of each perceptron define a single hyperplane which effectively divides the input space. This introduces a key limitation: a single perceptron can only model linear relationships. Each region of the input space on one side of the hyperplane maps to a specific output. Therefore, any problem where the classes cannot be separated by a single hyperplane cannot be solved by a single perceptron.

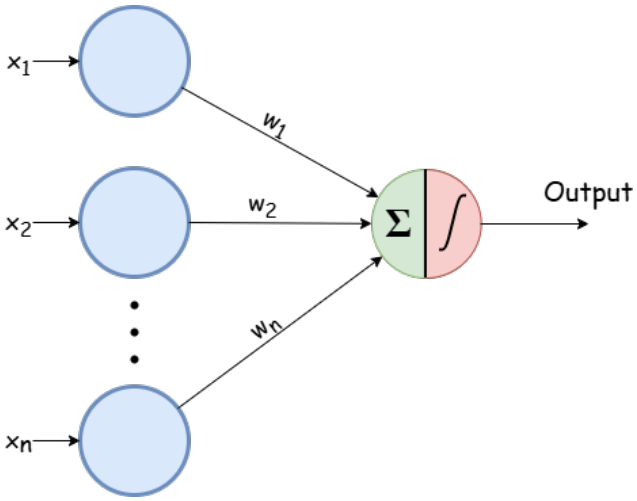


Figure 4.2.1: A single Neuron (Perceptron) [24].

To overcome this limitation, neurons are stacked into layers, forming a Multilayer Perceptron (MLP). By combining multiple layers and nonlinear activation functions, multilayer perceptrons can model more complex and nonlinear relationships. In a multilayer perceptron with L layers, the output of layer l is:

$$\mathbf{h}^{(l)} = \phi\Big(\mathbf{W}^{(l)T}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\Big) \tag{4.2.2}$$

Where $\mathbf{h}^{(l-1)}$ is the output of the previous layer and $\mathbf{h}^{(0)} = \mathbf{x}$ is the input vector.

In music applications, multilayer perceptrons have been applied to tasks such as chord recognition and instrument classification. Deep neural networks are used to transform chroma features into more discriminative representations for chord recognition [25]. Similarly, multilayer perceptrons and convolutional architectures are applied to predominant instrument recognition in polyphonic audio, showing that neural networks can effectively model timbral patterns [26].

4.2.2 Activation Functions

Activation Functions determine how a neuron responds to its input and are essential for introducing nonlinearity into a neural network. Without nonlinear activation functions, using a linear activation function for example, the network behaves like a single-layer linear model and fails to capture complex relationships. A few examples of activation functions are shown below.

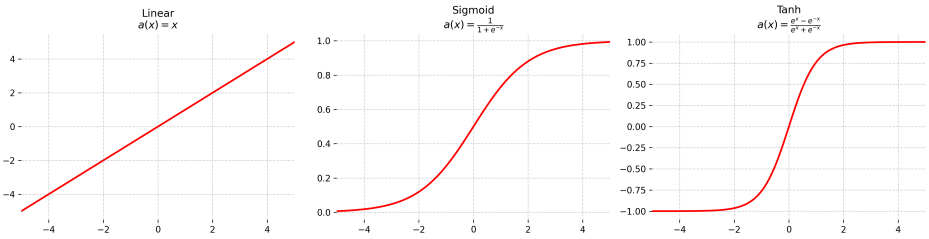


Figure 4.2.2: Linear, sigmoid and hyperbolic tangent activation functions.

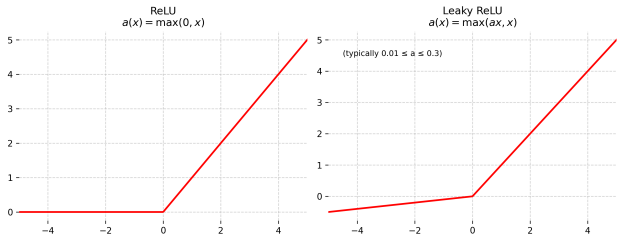


Figure 4.2.3: A single Neuron (Perceptron).

Nonlinear functions such as sigmoid, tanh and ReLU allow networks to approximate complex relationships, but they each have its limitations. Sigmoid and tanh can suffer from the vanishing gradient problem, which will be explained in Chapter 5, making training unstable. The ReLU can produce "dead" neurons when its inputs remain negative, "leaky" ReLU mitigates this problem [27], while softmax is typically limited to producing probability distribution in output layers.

4.2.3 Loss, Optimization and Backpropagation

To evaluate a model's performance, meaning how well its predictions match the output loss functions or cost functions are used. They provide a quantitative signal that the model uses to adjust its parameters during training. Common loss functions include:

Mean Squared Error (MSE): Better suited for regressions tasks, measures the average squared difference between predicted $\hat{y}_i$ and true y_i values over N samples.

$$L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{4.2.3}$$

Cross Entropy (CE): Measure the difference between predicted probability distributions and true class labels. Mostly used with classification tasks with C classes, where y_i is the true one-hot label and $\hat{y}_i$ is the predicted probability vector:

$$L_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\,\log\left(\hat{y}_{i,j}\right) \tag{4.2.4}$$

After calculating the loss, optimization algorithms make use of it and update the model's parameters accordingly trying to minimize it. The simplest approach is gradient descent, which moves each parameter θ in the direction of the negative gradient of the loss $\nabla_\theta L(\theta_t)$ at step t. The update rule is:

$$\theta_{t+1} = \theta_t - \eta\,\nabla_\theta L(\theta_t) \tag{4.2.5}$$

Where η is the learning rate which controls the step size. Choosing an appropriate learning rate is crucial, if too high it may cause divergence, if too low, training slows and might not reach optimal solutions. Variants of gradient descent that improve convergence are:

- **Stochastic Gradient Descent (SGD):** Avoids using the whole dataset for updates by using mini-batches instead.
- **Adam:** Combines momentum and adaptive learning rates for faster convergence.
- **RMSprop:** Adjusts learning rates based on recent gradient magnitudes.

Even though the process of calculating the gradients is conceptually simple, computing gradients for all parameters directly is computationally inefficient. To address this, neural networks employ backpropagation, an efficient algorithm that applies the chain rule of calculus to propagate the error from the output layer back through the network. This enables efficient gradient computation for all parameters during training.

4.3 Training and Generalization

Training a model typically requires multiple passes, or epochs, over the entire training dataset. Within each epoch the data are divided into batches, and the model processes one batch per step. Each step produces outputs with corresponding losses, which are then used to update the model's parameters.

The goal of the training process is to identify meaningful intrinsic patterns in the training dataset that will help the model generalize and apply this knowledge on unseen data to make decisions. This involves not only optimizing the model's parameters to minimize the loss of the training data but also ensuring that the learned patterns are transferable to unseen data. Generalization is therefore a central concern in model development and evaluation.

A common approach during model development is to split the dataset into three parts train, validation and test sets. Each set is selected in a way that ensures they are mutually exclusive, with the idea of training the model using the training set, using the validation set to make refinements and then evaluating the final model on the test set.

Different problems can occur during training that show lack of generalization, which are categorized based on the model's performance across these sets. Overfitting occurs when the model achieves reliable results on the training data but performs poorly on validation and test sets. This happens because the model memorizes noise or pseudo patterns in the training data rather than capturing their underlying structure. On the other hand, Underfitting occurs when a model, lacking sufficient capacity or training, fails to detect any meaningful patterns resulting in high error rates across all sets.

To avoid overfitting, regularization techniques introduce constraints that encourage more robust models, such as weight decay, which penalizes large parameter values, and data augmentation, which increases the diversity of training examples. Underfitting can be mitigated by increasing the model's complexity, extending training, or reducing regularization.

Two widely adopted methods for controlling overfitting are early stopping and dropout. Early stopping uses the validation set performance as an indicator and when its loss plateaus or starts to increase, it halts the optimization process, preventing the model from fitting to noise. Dropout attempts to prevent co-adaptation of features by deactivating a fraction of neurons at each step randomly. This way the model is encouraged to learn more generalizable representations instead of focusing on specific features.

4.4 Deep Learning

As mentioned above, stacking multiple layers to create artificial neural networks allows capturing more complex nonlinear relationships amongst the data. Deep Learning (DL) is a subset of machine learning that is based on multi-layer networks to extract hierarchical representations from data. The layers between the input and output layer are called hidden layers because they don't interact with the environment. Deep Learning models learn internal representations directly from raw input, without the need of handcrafted features, enabling them to distinguish more complex and meaningful patterns.

These models include a variety of architectures optimized for different types of data and tasks. Multilayer Perceptrons (MLPs) are fully connected networks where each neuron in one layer is connected to every other neuron on the next layer, forming the foundation of deep learning. Convolutional Neural Networks (CNNs) use convolutional filters to capture spatial hierarchies in data, making them best suited for grid-like inputs such as images or power grids. Recurrent Neural Networks (RNNs) maintain internal state information, acting like memory, to model sequential dependencies in data, while Transformer models use attention mechanisms to capture long-range relationships in sequencies processing them whole, avoiding recurrence. Transformers also contribute to Generative models.

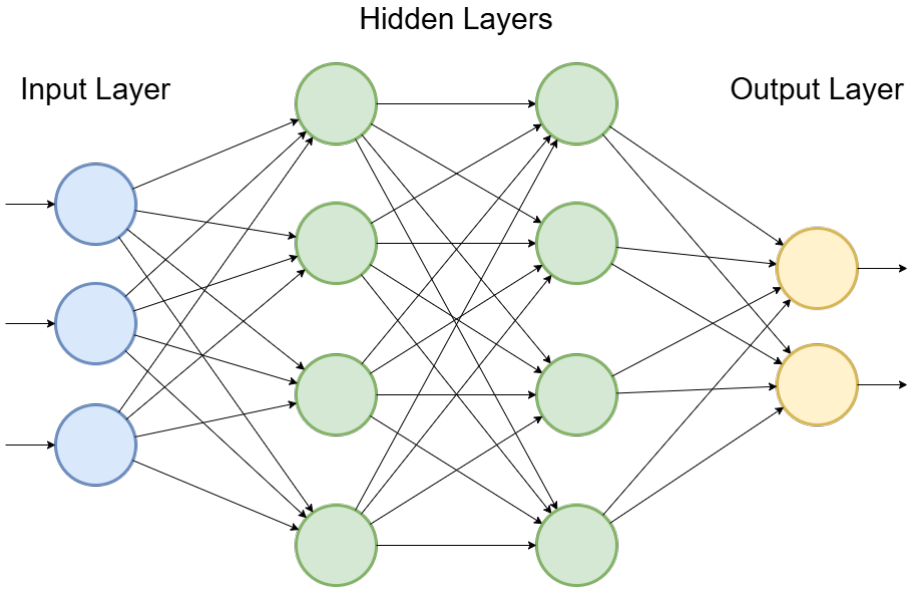


Figure 4.4.1: Multi-Layer Perceptron, 2 hidden layers, 4 units each.

Generative models use sophisticated methods to generate new data based on the data they were trained on. Other generative models are Generative Adversarial Networks (GANs) which consist of a generator that produces synthetic data and a discriminator that evaluates it and Diffusion models, that apply random noise on inputs and learn how to denoise that product and create new data.

Additional architectural implementations extend deep learning to more specialized data or tasks. Mamba models, based on Selective State Space Models (SSMs), efficiently model long sequences by maintaining a compact state representation, instead of relying on attention mechanisms, making them competitive alternatives to Transformers models for sequential tasks. Graph Neural Networks (GNNs) operate on data that can be represented as graphs and capture the relationships between them by propagating or aggregating information along the edges.

Training deep networks means optimizing a large number of parameters, typically though gradient descent, to minimize a loss function. Despite the flexibility and power of deep models this optimization comes with high computational costs, multiple layers present issues such as vanishing or exploding gradients and to make them more accurate large datasets are needed. To help mitigate those difficulties, techniques such as normalization layers, residual connections and advanced optimizers are applied.

Through the combination of different architectures with large-scale data, deep learning has dominated a wide range of applications. While this section introduces the architectures broadly, specific families such as Recurrent Neural Networks (RNNs), Long Short-Term Memory models (LSTMs), Gated Recurrent Unit models (GRUs) and Mamba models are discussed in detail in Chapter 4.

4.5 Tokenization, Embeddings and Feature Spaces

A central idea in machine learning is that raw data are not always suitable for learning, they might carry excessive information including noise. To mitigate this issue, algorithms for transforming the data into feature representations are used to produce structured numerical forms that capture the essential information. How well a model learns and generalizes is largely determined by the quality of the training data available and the way they are represented.

Before the surfacing of deep learning, feature extraction relied heavily on statistical and geometric techniques that aspired to reduce the complexity of the data and increase interpretability without losing meaningful information about them. One of the most widely used methods to reduce the dimensionality of the data is Principal Component Analysis (PCA).

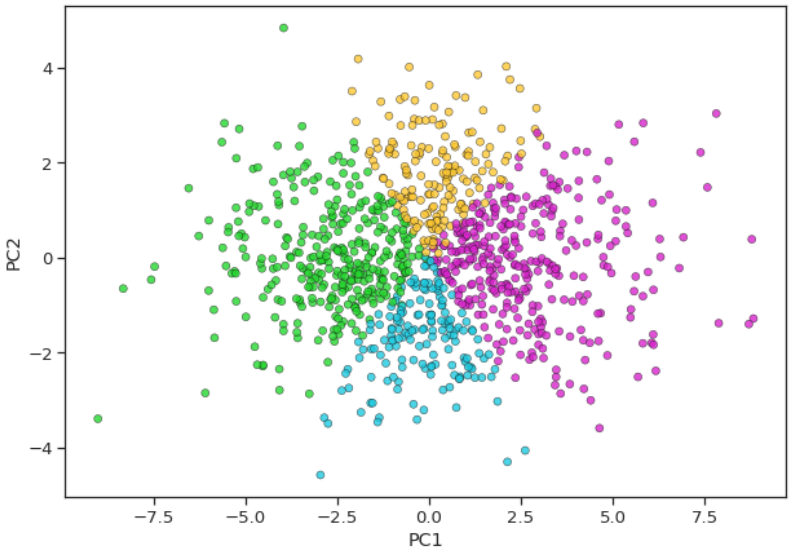


Figure 4.5.1: Principal Component Analysis in two dimensions, using two principal components [28].

Principal Component Analysis is an adaptive algorithm that finds new uncorrelated variables to represent the data, known as principal components, while retaining the variance of the original form. This not only makes learning more efficient but also provides a way to visualize high dimensional data into lower dimensions while helping reveal latent patterns that were indistinguishable before.

In contrast, Hierarchical Cluster Analysis (HCA) focuses on finding similarities between data points by creating hierarchies of clusters using two general strategies. Agglomerative clustering starts by having one cluster per data point and successively merges those clusters based on a distance metric or linkage criterion. Eventually, the process is terminated when all data points are combined into one single cluster, or some halting criterion is met. On the other hand, divisive clustering starts with one cluster for all the data points and recursively splits them into smaller clusters, one at a time, using a criterion like maximizing the distance

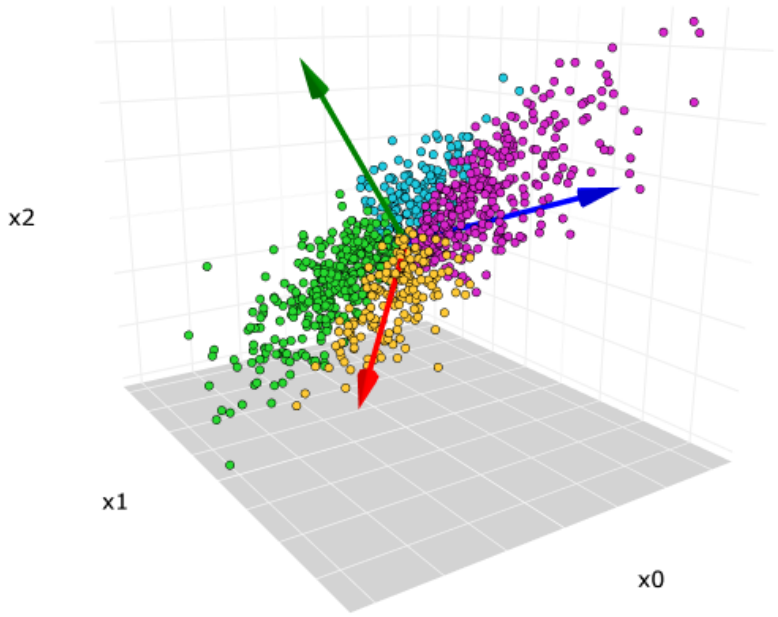


Figure 4.5.2: Principal Component Analysis in three dimensions, the colored arrows represent the new axis of the principal components [28].

that separates them. This representation creates a tree-like formation that allows analyzing the data at multiple levels of granularity, offering insights into the underlying structure of the dataset.

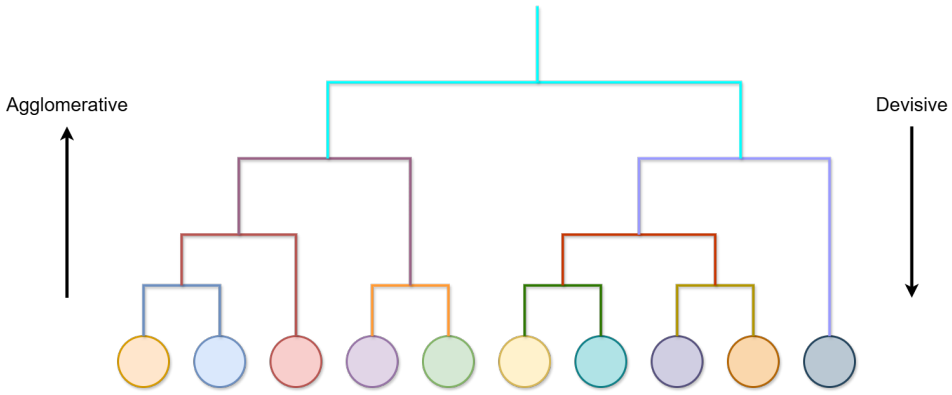


Figure 4.5.3: Hierarchical Clustering Dendogram.

These traditional methods laid the conceptual groundwork for the advent of modern representation learning. While they relied on mathematical criteria, deep learning methods use the data directly to learn these relationships. In modern neural models, the first step of this process is tokenization, where raw inputs, such as text, images, sound, or symbolic music, are replaced with meaningful units called tokens. Each token is then mapped to an embedding, a dense vector that represents the data numerically and carries information about semantic and contextual relationships.

This transition, from algorithmically defined feature spaces to learned embeddings, marks a significant shift in how models manage information. While methods such as principal component analysis and hierarchical cluster analysis are adaptive in the sense that they respond to the input data, their structure is constrained by fixed rules. As a result, these algorithms can only capture certain aspects of the data and miss complex,

nonlinear relationships. Learned embeddings, produced through neural networks, are free to roam in a continuous vector space, adapting their positions to capture semantic and contextual relationships is a task-driven manner.

Despite their flexibility and effectiveness, learned embeddings are not without limitations, they can be computationally demanding to train and store, they may overfit or inherit biases from the data making it difficult to transfer effectively between domains. Nevertheless, they provide a way for neural networks to automatically discover complex and task-specific patterns, creating rich features, a property desirable in applications such as those explored in this thesis.

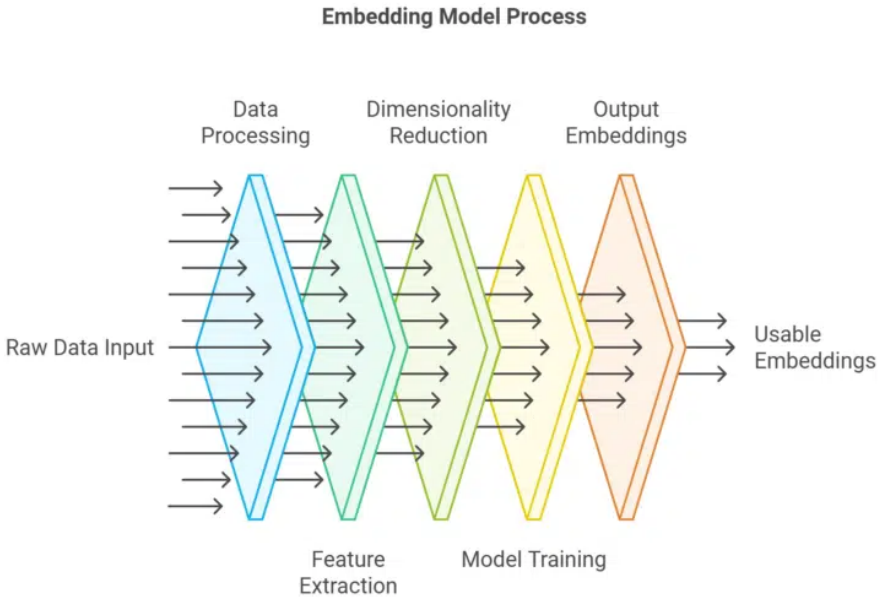


Figure 4.5.4: The embedding learning process [29].

Chapter 5

Deep Learning for Sequential Data

Music is fundamentally a temporal art form, the ordered succession of events, like notes, chords, and rhythmic accents, creates its identity. In contrast with static data domains, sequential structure carries essential musical knowledge. Concepts that rely on time dependent patterns like tension, resolution, and repetition are what shape harmony, melody and rhythm. These structural characteristics are what make systems suited for temporal modeling essential for a variety music information retrieval tasks.

The advent of deep learning models, capable of capturing temporal relationships directly from data, provided a way to model long-term dependencies that earlier statistical methods failed to express. These models can correlate events through time to make inferences by using memory mechanisms to maintain important temporal context. To achieve that traditional models, like vanilla recurrent neural networks, incorporate recurrence while some variants, such as long short-term memory networks and gated recurrent unit, employ specialized gates for better performance.

More recently, attention-based and state-space models have expanded sequence modeling by providing stronger long-range effectiveness. Transformer-based models replace recurrence with self-attention mechanisms, while Mamba integrates state-space dynamics with selective recurrence to maintain efficiency, both achieving parallel processing the sequential elements. These architectures are suitable approaches for modeling extended musical sequences and temporally aligned multi-modal data.

The purpose of this chapter is to present the main deep learning architectures for sequential modeling. Describing their mechanisms, limitations, and their utilization in music information retrieval tasks, provides a better understanding of how such algorithms can model, analyze, and generate music by leveraging their inherent sequential structure.

Contents

5.1 Recurrent Neural Networks 59
5.1.1 Vanilla Recurrent Neural Networks 59
5.1.2 Long Short-Term Memory Networks (LSTMs) 60
5.1.3 Gated Recurrent Units (GRUs) 61
5.2 Transformer-based models 62
5.2.1 Positional Embeddings 62
5.2.2 Self-Attention Mechanism 62
5.2.3 Multi-Head Attention 63
5.2.4 Encoder-Decoder Architecture 63
5.2.5 Position-Wise Feed-Forward Network 64
5.2.6 Limitations 64
5.3 State-Space Models and Mamba 64
5.3.1 Foundation State-Space Models 64
5.3.2 Modern Adaptation for Sequence Modeling 65
5.3.3 The Mamba architecture 66

5.4 Architecture Comparison **66**
5.5 Related Work **67**

5.1 Recurrent Neural Networks

Sequential data play an important role in multiple real-world domains, in many cases time is an essential dimension that cannot be ignored correlating the data. Such cases range from natural language processing and speech to financial series and music. Traditional feedforward networks treat inputs as independent samples, completely ignoring temporal dependencies.

Early attempts to capture time, treated it as an extra variable and used spatial representations for it [3]. What these methods failed to account was that they expected a fixed length of inputs, they were also unable to define how much buffering of the input was needed and at which point each sample should be examined but most importantly they couldn't distinguish the difference between absolute and relative temporal position.

Instead of treating time simply as a new dimension, capitalizing on the effect it has on data as they are processed helped create some sense of memory. A simple Recurrent Neural Network architecture was introduced [3] to formalize the idea of using temporal context as an input, addressing the challenge of modeling sequences over time. What these networks did differently, from simple feedforward networks, was how they treated the flow of information across the layers. A hidden state was proposed to carry information across time steps, allowing the network to remember prior inputs and use them to generate future outputs.

This design enables recurrent neural networks to incorporate both short-term and long-term dependencies, making them suitable for sequential analysis tasks in a wide range of applications [3].

5.1.1 Vanilla Recurrent Neural Networks

The basic form of Recurrent Neural Networks, often referred to as a vanilla version, consists of a hidden state that is updated recurrently at each time step using a combination of the hidden state at the previous step and the model's input. Formally, the hidden state h_t at a time step t can be expressed as:

$$h_t = \phi\big(W_x x_t + W_h h_{t-1} + b_t\big) \tag{5.1.1}$$

Where ϕ represents the nonlinear activation function, W_x and W_h are the weight matrices of the layer's input and the layer's previous hidden state $h_{(t-1)}$ respectively and b_t correspond to the bias vector. The output at each time step is typically derived from the hidden state through an additional output transformation, enabling the network to infer based on sequential information.

In the case of stacked recurrent layers, the hidden state of each lower layer acts as the input to the next layer above. Consequently, each layer's hidden state can be viewed as its output, while only the final layer is usually connected to the final output of the network. This hierarchical structure works similarly with multi-layer neural networks, allowing the higher layers to learn more abstract representations, in this case temporal ones.

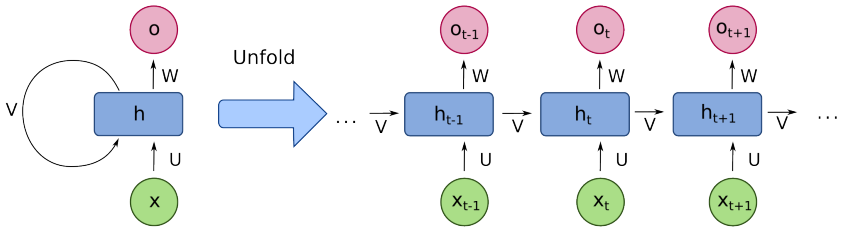


Figure 5.1.1: A recurrent neural unit and its unfold version [30].

Despite their conceptual simplicity, vanilla recurrent neural networks struggle to keep track of longer dependencies. During training on long sequences with Backpropagation Through Time (BPTT), gradients often vanish or explode. In a network with L layers, the gradient with respect to the activation layer l can be expressed as:

$$\frac{\partial L}{\partial h_l} = \frac{\partial L}{\partial h_L} \prod_{n=l+1}^{L} \frac{\partial h_n}{\partial h_{n-1}} \tag{5.1.2}$$

If the norm of each term $\frac{\partial h_n}{\partial h_{n-1}}$ is less than one, repeated multiplication causes the overall gradient to shrink exponentially as it propagates backwards through layers eventually vanishing. On the other hand, if the norm is greater than one the product grows uncontrollably producing the exploding gradient problem. Both effects lead to ineffective training in deep networks, since they make the feedback to early layers either negligible or unstable.

Recurrent neural networks can be viewed as deep networks unrolled over time, with one layer per time step. Because all these temporal layers share the same parameters, the gradient instability problem becomes even more noticeable and unavoidable as the sequences get longer, limiting them to short sequences [4].

5.1.2 Long Short-Term Memory Networks (LSTMs)

In search to surpass these limitations of the vanilla recurrent neural networks when faced with long sequences, the Long Short-Term Memory (LSTM) network was introduced [5]. This alternate architecture extends the standard recurrent neural networks by regulating the flow of information through a memory cell. To achieve this, it uses gating mechanisms and allows the network to selectively retain or forget information across long sequences. It splits the temporal information into a short-term part, the hidden state, and a long-term part, the cell state. Through this structure the gradient propagation becomes more stable, and the vanishing gradient problem gets mitigated.

Each long short-term memory cell is composed of three primary gates. The forget gate f_t, determines what fraction of the previous cell state $c_{(t-1)}$ should be preserved. The input gate i_t, controls the amount of new information that will be absorbed by the cell state from the input x_t. Lastly, the output gate o_t decides the amount of the cell state that is exposed for the new hidden state h_t to be calculated. All gates use a combination of the input x_t the previous hidden state $h_{(t-1)}$ and the previous cell state $c_{(t-1)}$. These operations, at each time step t, can be formalized as shown below:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f\right), \tag{5.1.3}$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i\right), \tag{5.1.4}$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o\right), \tag{5.1.5}$$

$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c\right), \tag{5.1.6}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{5.1.7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right). \tag{5.1.8}$$

Where σ denotes the sigmoid activation function and tanh is the hyperbolic tangent activation function, $\odot$ represents element-wise multiplication and b_f, b_i, b_o, b_c are the biases of forget, input, output and cell gates respectively. A common notation is also to represent the $Wx_t + Uh_{(t-1)}$ as a concatenation $W'[h_{(t-1)}, x_t]$, where W' is a combined weight matrix that stacks the weights W and U together.

These gating mechanisms allow the long short-term memory networks to maintain a constant error flow through the cell state c_t. The gradient can propagate backward through many time steps without vanishing or exploding because of the cell state update being a linear combination of its previous value and new candidate information. This stabilizes the learning process over long sequences and gives long short-term memory networks their advantage over vanilla recurrent neural networks. Even though they perform better, long short-term memory models are also accompanied by their own limitations. They have expanded the range on recurrent neural networks but are still underperforming on extremely long dependencies [32], [33]. They also require careful design choices, such as hidden size and learning rate selection, and are followed by significant memory consumption and computational complexity [34].

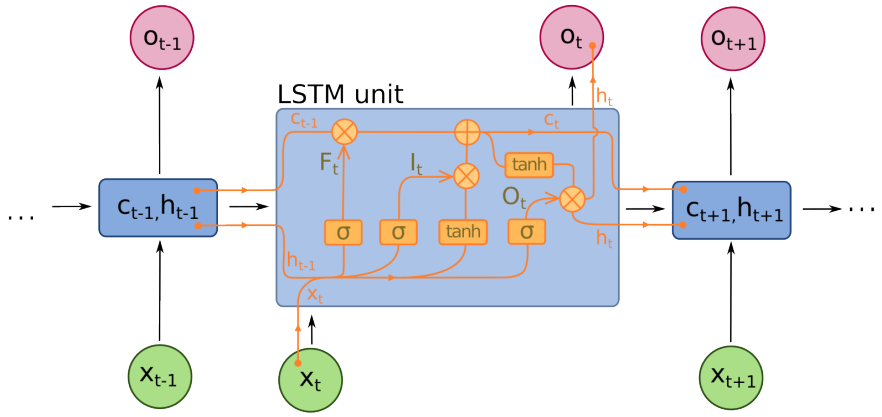


Figure 5.1.2: A Long Short-Term Memory unit [31].

5.1.3 Gated Recurrent Units (GRUs)

The Gated Recurrent Unit (GRU) [35] is a simplified variant of the long short-term memory cell with reduced architectural complexity that still retains the ability to capture long-term dependencies. What changes in the architecture is using a single gate, the update gate z_t to perform what the forget and input gates did on the long short-term memory cells and introducing the reset gate r_t to control the amount of influence the previous hidden state has to the candidate hidden state. By simplifying the model, the number of parameters is reduced while essential long-range dependencies still get detected, making gated recurrent units faster to train without significant loss of information.

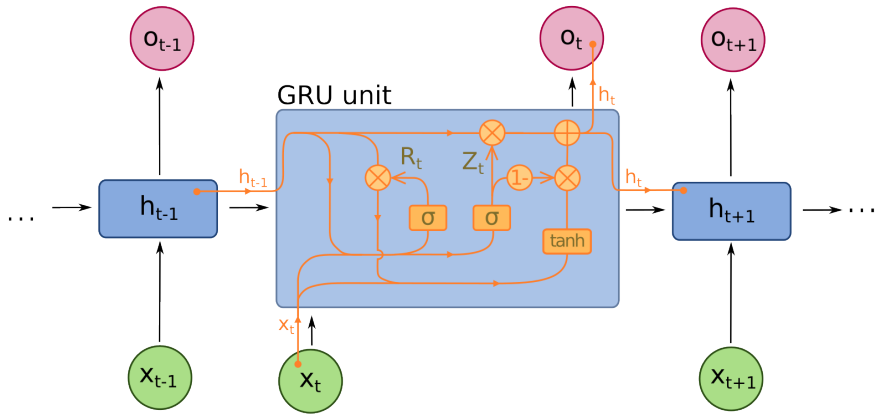


Figure 5.1.3: A Gated Recurrent Unit [36].

At each time step t, the gated recurrent unit updates its hidden state h_t according to the following equations:

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z\right), \tag{5.1.9}$$

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r\right), \tag{5.1.10}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \left(\mathbf{r}_t \odot \mathbf{h}_{t-1}\right) + \mathbf{b}_h\right), \tag{5.1.11}$$

$$\mathbf{h}_t = \left(\mathbf{1} - \mathbf{z}_t\right) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t. \tag{5.1.12}$$

Here, the update gate z_t balances how the previous hidden state $h_{(t-1)}$ and the candidate hidden state $\tilde{\mathbf{h}}_t$ influence the new hidden state, while the reset gate r_t determines how much the previous hidden state contributes to the candidate hidden state.

This gating mechanism enables the gated recurrent units to capture both short-term and long-term dependencies with comparable performance to long short-term memory units while simplifying the recurrent structure. Although their reduced complexity lowers computational and memory costs, allowing them to train faster, certain limitations still apply. Their sequential nature does not permit parallelization, and they can still struggle with extremely long dependencies and require careful tuning of hyperparameters like the rest recurrent architectures to achieve optimal performance [6].

5.2 Transformer-based models

The Transformer architecture [37] takes a different approach than recurrent neural networks in sequence modeling. It first eliminates recurrence, treating time as an extra dimension. While recurrent neural networks, long short-term memory units and gated recurrent units relied on sequential processing to capture temporal dependencies, with each time step depending on the previous hidden state, Transformers prioritize parallelization. They rely on different methods to counterbalance the lack of recurrence like positional encoding and self-attention mechanisms. These mechanisms try to combine positional information while embeddings are created from the input data and keep track of how they form relationships regardless of distance.

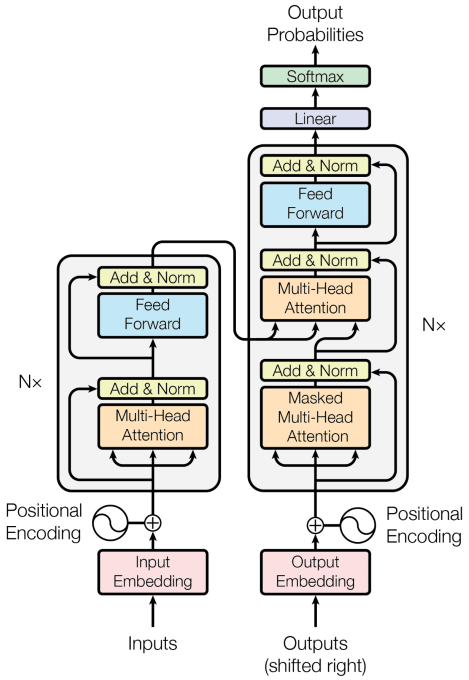


Figure 5.2.1: The transformer model architecture [37].

5.2.1 Positional Embeddings

Lacking the notion of sequence order, positional encodings are introduced to fill this gap in Transformer architecture. Either by using fixed sinusoidal functions or allowing the algorithm to learn them itself, positional embeddings are added to the input embeddings to help carry temporal information in the absence of recurrence.

5.2.2 Self-Attention Mechanism

The self-attention operation lies at the core of Transformer architecture. After the sequence is turned into embeddings the model learns three distinct representations for each element. These representations are the query (Q), the key (K) and value (V) vectors and they are derived simultaneously from the entire input sequence, enabling the parallel computation of them, unlike recurrent models that must process the input

sequentially. The attention mechanism computes how important each relationship of one element is with the others. To do this, it computes the similarity of its query with all the keys of the input and after applying softmax, for weight normalization, on them it combines the result with the corresponding values. This process in formally defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{5.2.1}$$

Where d_k is the dimension of the key vectors and is used to stabilize the gradients during training.

As an effect, context-aware representations are produced by combining the input features based on the resulting attention weights. This mechanism escapes the step-by-step propagation constraints that limited recurrent networks and enables the Transformer to capture dependencies over arbitrary distances, acquiring global context.

5.2.3 Multi-Head Attention

To take it a step further, Transformers also employ multi-head-attention, using several attention mechanisms in parallel. Each head learns to focus on different types of relationships, and their outputs are combined to produce the final attention output. This design helps analyze the sequence context on different levels within a single layer.

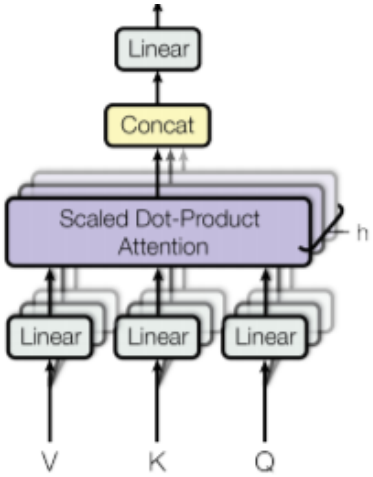


Figure 5.2.2: Multi-Head Attention: Several attention layers working in parallel [37].

5.2.4 Encoder-Decoder Architecture

The original Transformer model consists of two main components, an encoder and a decoder, each containing a stack of identical layers. The encoder takes the input sequence and transforms it into continuous representations that carry the contextual dependencies between elements. The decoder aims to create an output sequence combining the information of the previously generated tokens and the outputs of the encoder. Every encoder layer consists of two parts, the multi-head self-attention mechanism and a position wise feed-forward network, while the decoder layer also has an extra encoder-decoder attention sub-layer to account for the input from the encoder.

Additionally, each sub-layer is accompanied by residual connections and layer normalization to ensure stable gradient flow resulting in an efficient deep learning process. Generalization techniques and optimization methods are also applied.

5.2.5 Position-Wise Feed-Forward Network

The feed-forward network (FFN) is used to apply nonlinear transformations to the elements of the sequence. It uses the same parameters for every token, enabling once again parallelization maintaining computational efficiency. It is commonly structured as two linear layers separated by a nonlinear activation function as shown below:

$$\text{FFN}(\mathbf{x}) = \phi\big(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\big)\mathbf{W}_2 + \mathbf{b}_2 \tag{5.2.2}$$

Feed-forward networks work cooperatively with the attention mechanisms by refining the representations of each token individually, while the attention mechanism integrates contextual information across their positions. This combination allows each layer to learn complex position-wise representations and at the same time capture relationships between tokens.

5.2.6 Limitations

Despite their success, Transformers are faced with several limitations that surface when applied to real time tasks and long sequences.

The problem lies in the quadratic computation and memory requirements, $O(n^2)$, with respect to input sequence length of the model. This limitation occurs from the nature of the self-attention mechanism and affects the performance of the model. In addition, the lack of recurrence, even though it is partially balanced by the mechanisms capturing positional information, can lead to loss of information. Also, the advantage of being highly parallelizable comes with its own drawbacks, such as the need for extensive computational resources and large datasets for the training to be optimal.

Challenges such as these have motivated the exploration of less computationally demanding and less memory hungry architectures. Briefly, such architectures include linear attention transformers, recurrent-attention hybrids and state space models. These models attempt to absorb the modeling power of attention mechanisms and combine it with the temporal advantages of recurrent architectures. A promising recent development is the Mamba architecture, a state space model. Its aim is to offer the best of both worlds, the performance of Transformers with the efficiency of the recurrent networks. To achieve that it allows a form of structured recurrence without losing its parallelization advantages.

5.3 State-Space Models and Mamba

State-Space Models (SSMs), much like other recurrent networks, maintain a latent representation of the system's state at each given moment that evolves over, capturing temporal dependencies of sequential data and influencing the outputs. Originally adapted by fields based on recurrence such as control theory, signal processing and time series modeling, they recently entered the field of deep learning [38], [39], focused on long sequences of data. More specifically, the recent Mama architecture builds on this framework aiming for higher modeling capacity and improved efficiency.

5.3.1 Foundation State-Space Models

Like traditional recurrent architectures, the state-space models describe how the hidden internal state evolves over time according to the previous one and the new input and how the output is affected by the new hidden state and the new input. In their basic form these models rely on matrices A,B,C,D to update the information stored and produce its output. The formal representation of the equations is shown below:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t, \tag{5.3.1}$$
$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t. \tag{5.3.2}$$

The main difference from the recurrent models is the extra layer that combines the latent state and the input to derive the output instead of just using the hidden state and a fully connected network at the end. Here,

the x_t represents the latent state vector, u_t the input vector and y_t the output vector at time step t. In this simplified description, it is important to note that the latent state is what captures the context of past inputs removing the need for individual element attention.

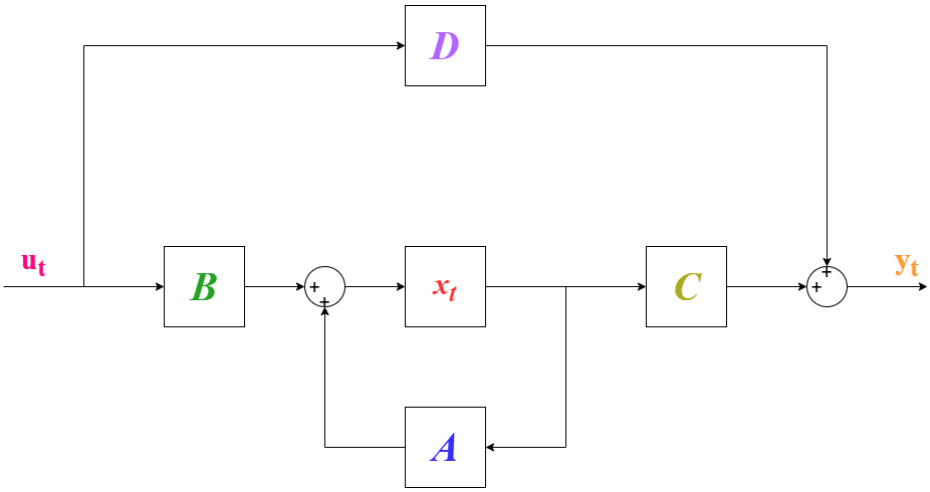


Figure 5.3.1: Representation of discrete time state-space model equations.

Concerning the deep learning sequence models, this formulation skips self-attention and relies on the latent state to summarize the past and while also updating with each new input.

5.3.2 Modern Adaptation for Sequence Modeling

Most recently, state-space models have been effectively utilized in deep learning sequential tasks. A variant of them, the S4 [38], is a structured state-space model that is based on a new parameterization of the state-space model having increased computational efficiency compared to previous approaches, while preserving their theoretical strengths.

In the discrete linear state-space model form, the output at time step t can also be expressed recursively by unrolling 5.3.2 as follows:

$$\mathbf{y}_t = \sum_{k=0}^{t} \mathbf{C}\mathbf{A}^{t-k}\mathbf{B}\mathbf{u}_k + \mathbf{D}\mathbf{u}_t \tag{5.3.3}$$

The filter derived from this formulation can be represented in convolution notation using $\mathbf{K} = \{\mathbf{CB},\ \mathbf{CAB},\ \ldots,\ \mathbf{CA}^k\mathbf{B},\ \ldots\}$ as a kernel:

$$\mathbf{y} = \mathbf{K}\mathbf{u} + \mathbf{D}\mathbf{u} \tag{5.3.4}$$

The S4 model leverages this structure by parameterizing system matrix A to allow for efficient diagonalization, making the computation of CA^kB elements both stable and parallelizable. This approach combines the modeling of long-range dependencies with linear computational complexity relative to the sequence length, as opposed to the quadratic cost associated with self-attention. As a result the S4 and its variants have equipped the state-space models with the practical scalability required for modern deep learning applications without losing their theoretical strengths, becoming a powerful tool in multiple domains that are based on sequential information such as language, audio and time series.

Despite its strengths, the S4 model is bounded by limitations. Its structured parameterization, of A, introduces stability and more efficient computation but may struggle when faced with complex or highly dynamic sequences. Also, even though it operates on linear time, longer inputs present large memory requirements resulting in a bottleneck. Finally, it tends to ignore local patterns and focuses more on global dependencies.

5.3.3 The Mamba architecture

The Mamba architecture [39] was proposed as an advancement of the S4 to address such challenges without losing the state-space character and its efficiency. It introduces a selective state-space mechanism, making the state-space parameters functions of the input, enabling dynamic decisions based on the content of the input and adapting to it. This procedure creates a result similar to the attention mechanism of Transformers but avoids the quadratic cost. In addition, Mamba, like Transformer's multi-head attention effect, exhibits multi-scale behavior, focusing on varying temporal resolutions thanks to its selective mechanism, capturing dependencies at variable ranges. Finally, its architecture allows it to scale linearly with sequence length, maintaining constant memory per token and taking advantage of hardware-aware parallel methods to optimize computations, effectively addressing memory bottlenecks.

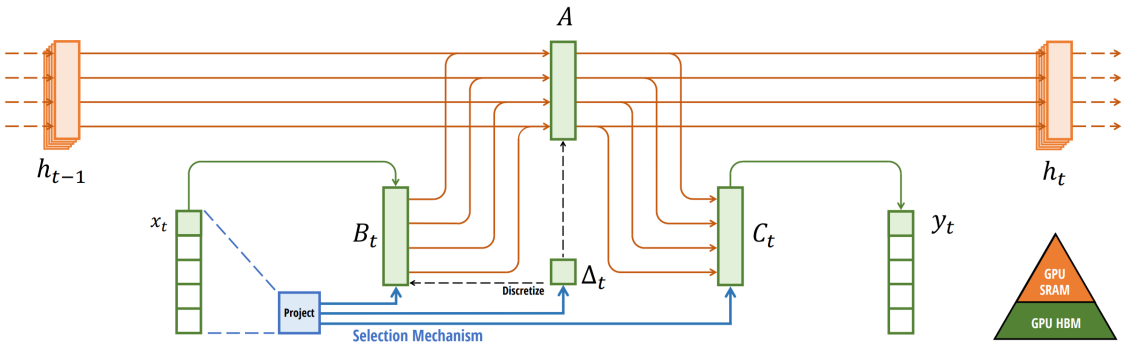


Figure 5.3.2: The mamba model architecture including selective memory allocation for efficient computing [39].

Even though Mamba is an upgrade of the S4 model, trade-offs still exist. The selective mechanism complicates each step's process, and its multi-scale behavior makes the model more complex and harder to interpret. It also relies heavily on hardware-aware implementations to optimize its performance, while its increased flexibility can lead to inefficiencies with short sequences or tasks with limited data.

5.4 Architecture Comparison

To summarize, sequential models have evolved to handle both short and long-range dependencies while maintaining computational efficiency. The vanilla recurrent neural networks introduce a hidden state that updates recurrently with every input, but struggle on longer sequences because of the problems of vanishing and exploding gradients. Architectures such as long short-term memory networks and gated recurrent units incorporate gated mechanisms to mitigate these issues and improve stability and performance on moderately long sequences, while still struggling with extremely longer ones. By being inherently sequential, these models lose on training and inference by their inability to utilize parallelization techniques.

Transformers abandon recurrence in favor of parallelization and make up for it by introducing the self-attention mechanism and embedding positional information into the representation of the data. They achieve high performance capturing both local and global relationships while being flexible enough but their quadratic complexity with respect to sequence length marks them as memory hungry and less efficient when faced with very long sequences.

Mamba architecture represents the state-of-art approach of state-space models, combining the efficiency of recurrent methods with the parallelizable aspect of transformers. Its selective state-space mechanism allows it to capture information about the input content dynamically while also focusing both on short and long-term dependencies. Long sequences do not pose much of a restriction as Mamba models scale linearly with sequence length. On the other hand, it might underperform to shorter sequences because the large number of parameters per token can lead to overfitting and the overhead added by the gating mechanisms can stall performance. In this case simpler models like long short-term memory networks and gated recurrent units may be more efficient.

5.5 Related Work

The development of deep learning models for music information retrieval purposes has seen significant progress in recent years as sequence modeling architectures become more capable of capturing long-term temporal dependencies in music. From more traditional recurrent models to attention-based ones and state space models, this section provides some examples of their usage in music information retrieval tasks.

Important studies in the area of music generations based on recurrent networks. It was demonstrated by Eck and Schmidhuber in [40] how long short-term memory systems can learn temporal dependencies in blues music compositions and generate analogous coherent musical progressions. They showed that once the network found the relevant structure it stuck to it without drifting away. In [41], Garoufis et al. proposed a long short-term memory network that utilized neural attention mechanisms to learn mappings between symbolic representations of polyphonic chord progressions and future chord candidates.

Gated recurrent unit architectures have also been applied to music information retrieval tasks. As explained by Han et al. in [42], a combination of a residual convolutional neural network and a gated recurrent unit was used as a basis for music emotion recognition. This hybrid architecture shows how recurrent networks can be efficiently implemented for modeling sequential features, while preserving local feature correlations.

With the upcoming of attention mechanisms, Transformer models were also introduced in sequence modeling for music information retrieval tasks. As shown in [43] by Huang et al., these architectures were incorporated to create a music Transformer capable of creating minute-long compositions with compelling structure by leveraging relative attention mechanisms. An extension of this approach was developed in [44] by Huang and Yang with the pop music transformer, which makes use of beat-based event representations to model rhythmic structure in piano compositions taking music generation a step further.

At the same time, the use of state space models for sequence modeling has proven to be efficient in capturing long-term dependencies in music while maintaining computational efficiency. A state space model was presented in [45] by Krebs et al. for joint tempo and meter tracking. Approaching tempo estimation as a dynamic state space problem lead to accurately modeling hierarchical rhythmic structure and reduced computational complexity. More recently, Tang et al. developed a Mamba based architecture for music-guided dance video synthesis [46]. The model integrates Mamba blocks as a way to capture both spatial dependencies between joints and long-range temporal dependencies. This enables the generation of coherent dance sequences following the flow of a specific musical piece, uncovering a new path in multi-modal music information retrieval tasks, by modeling complex musical sequences beyond symbolic and audio representations.

Other, multi-modal approaches in this field have also adapted deep sequence models to process multiple information sources. In [47], Pyrovolakis et al. used transformer-based models alongside long short-term memory networks. They examined the difference in inference between using audio signals and lyrics separately as inputs or using them both for a multi-modal analysis. The results proved that multi-modal representations outperform single-modality models.

Sequence modeling has become a central piece across a variety of music information retrieval tasks. From chord prediction and emotion recognition to expressive music generation, sequential models have demonstrated how temporal and contextual dependencies are crucial to music information retrieval systems. Whether through recurrent mechanisms, attention-based methods, or state-space architectures, the ability to model sequential relationships allows systems to capture the complex structures in music, making these models essential for modern music information retrieval tasks.

Chapter 6

Latent Distance Models

Social networks are complex systems in which entities, represented by nodes, are connected with edges, much like a graph, whose purpose is to be descriptive of the nature of their interactions. These networks are applicable in various domains, from modeling human relationships, like friendships or professional collaborations, to representing biological interactions and information flow. Analyzing these networks gives essential insight into their structure, dynamics, and the significant principles that surround them. Traditional network models often rely on explicitly described relational features or simple observed covariates. However, many real-world paradigms exhibit revealing patterns indirectly. For example, in social networks that embed human friendships, individuals may form connections based on latent characteristics, them being personality traits, shared interests or unnoticed social contexts.

Latent Distance Models (LDMs) provide a structured approach to reveal the hidden patterns on which connections are established. These models assume that a network can be represented in a latent space, typically Euclidean, with each node occupying a position on it [7], [10]. Instead of explicit connections between two nodes, their distance in the latent space determines the probability of them being related. Nodes closer together in this space are more likely to form an edge than those further apart. Latent distance models provide a geometric perspective that combines interpretability and flexibility while also capturing complex network structures like transitivity, clustering, and homophily without the need for manual feature selection for each structural pattern.

Aside from social networks, latent distance models can be adapted to model different kinds of relationships that could be conceptualized in the form of a network. Musical elements such as chords, motifs, or n-grams can be represented as nodes in a relational network, with edges absorbing information about co-occurrence or transition likelihoods. By embedding such elements in a latent space, implicit similarities between musical events may surface that are not visible from a shallow level analysis. While initial attempts applied to n-gram based chord progressions haven't shown strong predictive performance, this framework still offers valuable insights about the structural and relational patterns inherent in musical compositions.

In the following sections, the theory and methodology of latent distance models will be presented, alongside their extensions and generalizations. Also, inference and estimation strategies will be discussed as well as potential applications in the domain of music. A thorough understanding of these models provides insights as to why more complex sequence models are needed but also highlights their interpretive power.

Contents

6.1 Preliminaries **71**
6.1.1 Logistic Regression 71
6.1.2 Maximum Likelihood Estimation (MLE) 71
6.1.3 Bayesian Methods 71
6.1.4 Markov Chain Monte Carlo (MCMC) 71
6.2 The Latent Distance Model (LDM) **72**
6.3 Variants and Extensions **73**

6.4 Limitations **73**

6.1 Preliminaries

Latent distance models, like many probabilistic models, rely on statistical methods for parameter and latent variable estimation. Methods mentioned in this chapter are the statistical ones explained below.

6.1.1 Logistic Regression

Logistic regression models are statistical models for binary outcomes. They define how features x influence the outcome of an event y using the logistic function:

$$\Pr(y = 1 \mid x) = \frac{1}{1 + e^{-x^\top \beta}} \tag{6.1.1}$$

With β being the model's parameters. The log-odds of the event are linear with respect to the features:

$$\log\left(\frac{\Pr(y = 1 \mid x)}{\Pr(y = 0 \mid x)}\right) = x^\top \beta \tag{6.1.2}$$

In latent distance models, this function determines edge existence probability based on distances in the latent space.

6.1.2 Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation is a parameter estimation method that decides based on which parameter values maximize the likelihood of the observed data:

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} L(\theta; \text{data}) \tag{6.1.3}$$

Where $L(\theta; \text{data}) = \Pr(\text{data} \mid \theta)$.

In latent distance models, maximum likelihood estimation can estimate latent position z_i and parameters like β that make the observed network most likely to have occurred under the model.

6.1.3 Bayesian Methods

Bayesian inference treats parameters as random variables that have prior distributions $\mathrm{p}(\vartheta)$ and uses the observed data to form the posterior distribution:

$$\Pr(\theta \mid \text{data}) \propto \Pr(\text{data} \mid \theta)\ \Pr(\theta) \tag{6.1.4}$$

Where $\Pr(\text{data} \mid \theta)$ is the likelihood of the data based on the parameters θ.

In contrast with maximum likelihood estimation, which specifies the parameter values, Bayesian methods provide a distribution over the parameters.

6.1.4 Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo is a computational method for sampling from a complex probability distribution, such as the posterior Bayesian models. It constructs a sequence of dependent samples approximating the target distribution over time.

6.2 The Latent Distance Model (LDM)

Latent Distance Models represent networks by mapping each node i to vector z_i in a latent Euclidean space of dimension $d, z_i \in \mathbb{R}^d$. The idea behind this model is that the distance separating two nodes reflects the probability of an edge existing between them. Nodes closer together are more probable to share a connection than those further apart.

For the undirected version of this model the probability of an edge between nodes i and j is modeled using a logistic link function [7]:

$$\Pr(Y_{i,j} = 1 \mid z_i, z_j, \beta) = \frac{1}{1 + e^{\|z_i - z_j\| - \beta}} \tag{6.2.1}$$

Or in logit form:

$$\text{logit } \Pr(Y_{i,j} = 1 \mid z_i, z_j, \beta) = \beta - \|z_i - z_j\| \tag{6.2.2}$$

Above, $Y_{i,j}$ is the adjacency matrix, $Y_{i,j} = 1$ indicates edge existence and $Y_{i,j} = 0$ the opposite, $\|z_i - z_j\|$ denotes the Euclidean distance between nodes i and j and β is the global parameter that controls how dense the latent space gets. Smaller β values decrease the probability of an edge for a given distance, while larger ones increase it, effectively shifting the logistic curve and indirectly constraining the geometry so the model can fit to the data.

Proximity in the latent space hints at conceptually of functionally similar nodes. This structure allows network properties such as transitivity, clustering and homophily to be expressed and captured by the model, with no need for explicit feature encoding. Directed networks can also be implemented by assigning separate latent vectors for sending and receiving nodes, modifying the distance term to accommodate edge direction.

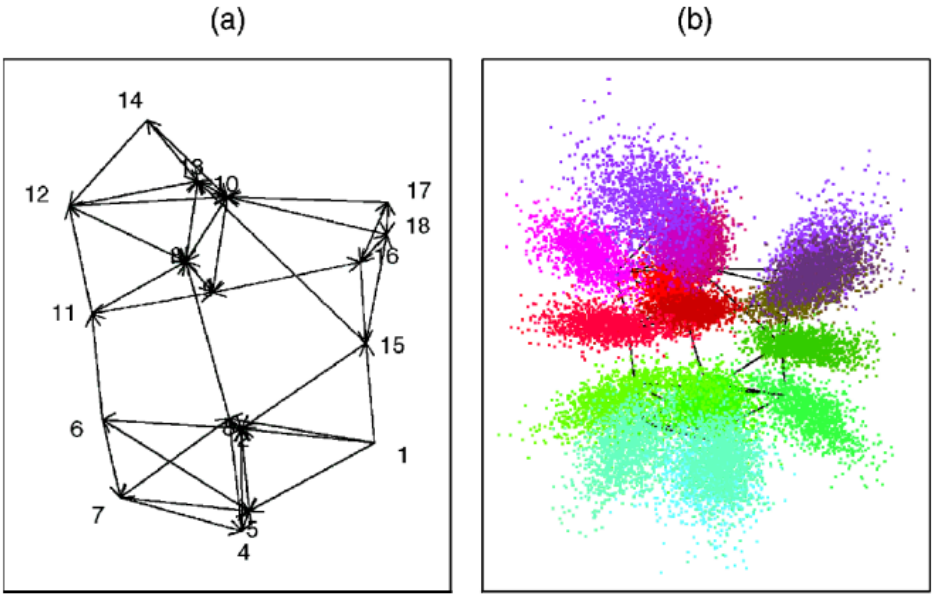


Figure 6.2.1: Latent Distance Model representation (a) using maximum likelihood estimation, (b) using Bayesian methods [7].

The parameters of the model, the latent positions z_i and β, are inferred from the observed adjacency matrix using statistical inference methods. Maximum likelihood estimation gives a single configuration that maximizes the likelihood of the observed network, while Bayesian inference, combined with Markov Chain Monte Carlo sampling, generates a distribution over the latent positions and parameters, inserting uncertainty to the model making it more flexible.

6.3 Variants and Extensions

Although the basic latent distance model provides a simple and interpretable way to represent relational data, several extensions exist. Variants include those that incorporate community structure by clustering latent positions [8], or those that allow temporal dynamics that account for time as a variable that influences relationships [9]. While time passes some relationships might emerge while others might fade. Other models adapt covariates to account for additional node or edge features that are observed, effectively improving predictive accuracy [10]. Alternative geometries have also been explored in place of the Euclidean space, such as hyperbolic or spherical latent spaces, allowing for better representation of scale-free or hierarchical networks [11].

6.4 Limitations

Latent Distance Models provide a flexible and interpretable framework for representing relational data by embedding nodes in a latent space. The distances between the latent positions determine the edge probabilities and allow the model to capture structural patterns.

Despite their strengths, their predictive performance is limited for sequence modeling. Defining nodes and edges, from representations that do not form them naturally, requires manual choices. They are also sensitive to latent space dimensionality, while the scaling parameter β can strongly influence results. In spite of these difficulties, latent distance models can offer valuable insights and provide a conceptual foundation, motivating the adoption of more well-suited sequence models.

Chapter 7

Proposal

7.1 Motivation and Research Objectives

The previous chapters have established the theoretical and computational foundations for sequence modeling, and the role of deep learning architectures and latent-space representations. Concerning musical applications, despite the rapid progress in the field of music information retrieval, most recent approaches are centered around more complex architectures, like transformer-based ones, that are accompanied by high computational costs and are presented like black boxes, providing little interpretation of their inference. Furthermore, approaches based on symbolic representations typically treat musical pieces and their chord progressions at the level of entire pieces, neglecting the internal part structure and compositional context that may shift across song sections.

This thesis revisits these challenges from a different perspective by taking advantage of simpler, more interpretable sequence models trained on a large-scale symbolic dataset of chord progressions, the Chordonomicon. A primary motivation is to investigate whether lightweight architectures, parameter wise and computationally wise, such as recurrent neural networks and state-space models can achieve competitive performance on symbolic music information retrieval tasks. At the same time, this study focuses on interpretability through analysis of the dataset and the use of latent distance models.

A key component of this study is leveraging the inherent structure of chords. In addition to their complete form, decomposing chords into their constituent parts, root, quality and bass, allows models to capture harmonic relationships separately between components, allowing for finer-grained analysis and explainability. Additionally, predictive inferences are made at the level of individual parts rather than entire songs, incorporating into the analysis the way the musical context changes across different sections of a composition. This part-based approach addresses a gap in the literature, as most prior work ignores the internal segmentations of musical pieces, assuming the context remains the same across them.

In the initial stage of this thesis, statistical methods based on n-grams are applied, providing statistically interpretable perspectives on musical structure in combination with latent distance models. By creating a latent space in which distances correspond to n-gram similarity, according to specific design criteria, the latent distance models lay out a way to uncover potentially musically meaningful patterns for predictive tasks such as genre and decade classification. Following this, experiments employing sequential models based on recurrence are conducted for the chord prediction task, utilizing decomposed chord representations and part-level contextual information.

Guided by these motivations, the main objectives of this thesis are as follows:

- Perform a statistical analysis of the large-scale symbolic dataset to identify evidence of musically meaningful structure.
- Apply interpretable models, such as latent distance models, to uncover harmonic and stylistic relationships based on n-grams statistics.

- Evaluate lightweight architectures, including recurrent and state-space models, on symbolic music information retrieval tasks.
- Explore decomposed chord representations, root, quality+extensions, bass, and part-level segmentation to assess their impact on predictive tasks.
- Compare results across modeling options, integrating findings from statistical, recurrent, and state-space approaches.

7.2 Overview of Proposed Framework

The experiments conducted in this thesis are split into two complementary methodological pipelines both sourcing their data from the Chordonomicon dataset, a large-scale collection of symbolic chord progressions supplemented with meta-data about part, genre, and decade. All analyses rely exclusively on symbolic representations to extract structural and harmonic properties rather than other modalities such as audio features.

Statistical and Latent-Space modeling for Classification

The first pipeline investigates interpretable representations of musical structure. Based on n-grams, statistical methods are employed to capture local harmonic dependencies, creating a foundation for further analysis of regularities found in the data. Latent distance models are used to embed chord progressions into a continuous latent space, where distances encode similarities between n-grams, enabling the identification of important patterns that carry information about musical structure, supporting tasks like genre and decade classification.

To prevent the combinatorial explosion following the unique n-grams, atonal representation for the latent distance approach is chosen. In this representation, each chord is abstracted, retaining only its root part and encoding it as a pitch-class number (0-11). This simplifies the representations while also preserving essential harmonic information. Furthermore, as a baseline to measure the performance of the latent distance representations on classification tasks, lightweight recurrent architectures are applied, allowing comparison between interpretable mappings and sequential learning.

Sequential Modeling for Chord Prediction

The second pipeline 7.2.1 is focused on predictive modeling of chord progressions. The task of chord prediction is explored through a variety of sequential architectures, including vanilla recurrent neural networks, long short-term memory networks, gated recurrent unit networks and mamba models. These models are employed to learn temporal dependencies and harmonic relationships in chord progressions. In contrast with the classification pipeline, this approach implements no abstractions and relies on full chord information, either by representing each chord as a single token or in a decomposed form by using three tokens that represent root, quality and bass.

Predictions are performed at the part level, enabling analysis of how musical context varies across song sections. This approach addresses a gap in prior work, where chord prediction is typically applied at the whole-song level, ignoring the contextual changes between parts. The combination of the part-level segmentation and chord decomposition representations aims to uncover musically meaningful patterns in harmonic sequences and improve predictive performance.

Overall Perspective

Together these two pipelines provide complementary perspectives. The statistical and latent space model-based pipeline focuses on exploring the dataset and its structure to provide insights into the relationship between the data. The sequential modeling pipeline dives into the predictive dimension and investigates chords sequences and how treating each chord and song as a construct composed of smaller pieces influences the modeling of harmonic sequences.

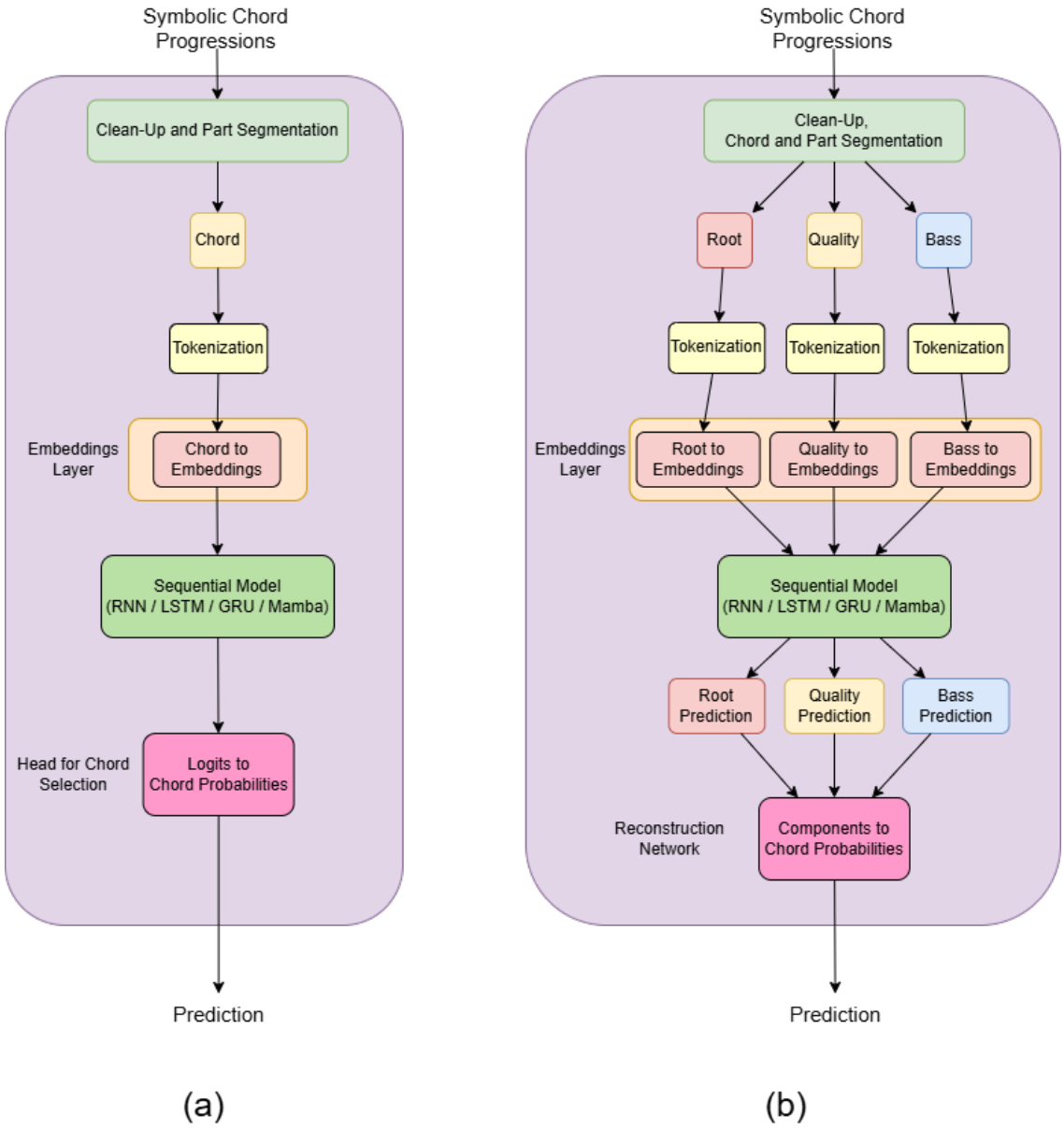


Figure 7.2.1: The two pipelines for the chord prediction models, (a) the one token per chord approach and (b) the one token per chord component approach (3-tokens).

By integrating these methods, this thesis seeks to balance explainable symbolic analysis with data-driven predictive modeling, demonstrating that computationally efficient and interpretable approaches can provide meaningful insights into the structure of tonal music. These conceptual frameworks provide the foundation for the analyses, experiments and results presented in the following chapters.

Chapter 8

Dataset and Analysis

Chord progressions constitute the essence of a piece of music, serving as the framework through which musicians create harmonic flow and coherence within their pieces. In recent years, Music Information Retrieval (MIR) has been a rapidly evolving field, with the goal to extract important information from music data and given their central role, analyzing music through chord progressions has become a key focus.

A crucial factor for advancing such analysis is the availability of well-structured datasets. While the development of such datasets has been rapid in other fields, music datasets have evolved more slowly due to a combination of challenges. Strict copyright laws, unavailability of audio sources, distribution shifts and the inherent ambiguity of music in its representation and labeling are some of the problems that present an obstacle to creating such datasets.

One promising resource, addressing these challenges, is the Chordonomicon dataset, which constitutes the largest collection of chord progressions to date. The dataset uses the standardized Harte Syntax for symbolic representation, ensuring consistency and compatibility in music analysis. Its extensive metadata further enhances its value, offering insights into aspects like genre, sub-genre, release dates and part segmentation info.

Contents

8.1 Dataset Overview 80
8.2 Feature Extraction 80
8.3 Descriptive analysis 81
8.3.1 Structural Features 81
8.3.2 Sequential Features 87
8.4 LDM analysis 90
8.4.1 Dataset Balancing 90
8.4.2 Chord Processing and N-gram Extraction 90
8.4.3 Category Graph 91
8.4.4 N-gram Graph 91
8.4.5 Node Coloring and mapping 91
8.4.6 Latent Distance Modeling 92

8.1 Dataset Overview

The Chordonomicon dataset is a collection of more than 666,000 annotations (approximately 679,000) of different songs paired with metadata. The chord progressions data were extracted from the Ultimate Guitar (UG) platform, where users contribute and rate chord transcription. In case of multiple submissions for a song, a weighted average of the user ratings and vote counts is calculated to determine the best version of a track that is also the one retained.

After extraction, a normalization procedure is applied to the corpus, essential for addressing its heterogeneous nature because of the user-entered annotations. Songs with less than four chords or bizarre chord encodings are filtered out and identical repeating chords in entries are collapsed into one (A G G A→ A G A). The last step ensures that the sequences reflect harmonic transitions rather than temporal repetitions. The product of this process is also curated by music experts, and by retaining only the chord progressions with high confidence, it is accurately converted into Harte syntax [12] and the syntax proposed by the Functional Harmony Ontology [13], ensuring musically meaningful and understandable representations of progressions.

Alongside the chord progressions, the dataset provides structural part labels corresponding to eight distinct categories: Intro, Verse, Chorus, Bridge, Interlude, Solo, Instrumental and Outro. The labeling splits songs into segments and aligns with previous works described in the Chordonomicon paper [14] on similar music information retrieval tasks. In case of chord progressions with unclassifiable parts, the labels from the whole song are ignored while the progressions are maintained.

The dataset is further enriched with metadata obtained with the usage of the Spotify Web Application Programming Interface (API). The metadata was collected after mapping entries based on track and artist names with the corresponding Spotify data, to ensure correctness. Each song, when data is available, is accompanied by release date, encoded as decade, main genre of the artist, and a subgenre of the track, and the Spotify identifiers of the song and the artists. The main genres consist of the twelve most common ones, that also cover other subgenres. Tracks whose artist's main genre isn't included in the top twelve do not have one assigned to them. Considering the different decades in the dataset they range from 1890s to 2020s.

Despite the large size of the dataset and the importance and variety of the information it provides, some limitations remain. Also mentioned in the Chordonomicon paper [14], source bias and noise from the user's annotations follows the data and some abstractions or errors may remain even after careful curation. Another problem lies in the imbalance of the data with respect to genre and the absence of some metadata. Nonetheless, the dataset provides extensive metadata and structural annotations while at the same time its size helps mitigate occurring inconsistencies.

8.2 Feature Extraction

The Chordonomicon is accompanied by an initial Exploratory Data Analysis (EDA), conducted by the creators (Kantarelis et al.) [14] to provide an overview of chord distribution and structural properties. Building on this foundation we conducted our own exploratory data analysis to extract some key features from the dataset and to provide more insights into its structure and the relationships between the data. To better understand the dataset, we converted the chord sequences from the Chordonomicon into several feature formats, each designed for a specific type of analysis.

Structural Song Features

At the song level, metadata about genre, decade of release and artist identifiers, were used to record distributions of songs and artists across categories. Considering the part-level, we extracted structural measures such as the number of parts per song and part sizes overall and with respect to genre and decade. These features do not depend on chord identity but support structural analyses.

Decomposition Features

The syntax of the dataset allows for easy decomposition of each chord to its canonical components, root, quality, extensions and bass note. This representation enables analysis of harmonic complexity and stylistic

usage by focusing separately on each part of a chord. Such analysis involves how qualities, extensions and bass are used across different genres and decades providing insights into stylistic and harmonical decisions on musical composition. Additionally, root information can be leveraged to study harmonical flow through root motions in semitones. Each root is represented by a number based on its position on the chromatic scale (0-11). By turning each root into its atonal representation, information about absolute key is ignored to focus on relative harmonic context and root-to-root relationships.

Full chord Features

For harmonic vocabulary analysis, each chord is retained in full form, except that explicit bass (inversion) information was omitted to focus on chord-type diversity rather than voicing choices. Statistics about the number of unique chords per decade and genre are gathered along with the most common bigrams and trigrams across all songs. Here the part structure is also taken into consideration by also considering n-grams that are split between part boundaries. For cross-part statistics we explicitly record n-grams whose last chord is also the first chord of the following part, enabling the study of patterns between song sections.

The latent distance model approach incorporates atonal root representations for each chord, to avoid the combinatorial complexity of n-grams as n grows. The modeling focused on n-grams of length 4,5 and 6, that were extracted from the dataset. To reduce category bias, the n-grams were sampled from songs after balancing the dataset with respect to genre and decade. Each n-gram was treated as a graph node, with edges defined via nearest neighbors in feature space. The resulting graphs formed the inputs for the latent distance models, producing latent positions in a 2-dimensional space. This process is described thoroughly in section 8.4.

8.3 Descriptive analysis

Building on the feature extraction described above, in this section we present the results of our Exploratory Data Analysis (EDA) on the Chordonomicon dataset. The analysis focuses on both static and dynamic features of the datasets, providing insights into structural properties and sequential features.

Structural properties provide information on the organization of songs and parts, as well as the harmonic vocabulary, including chord qualities, extensions, and unique chord usage. In addition, sequential features capture harmonic movement and relationships between neighboring elements, including root-motion intervals and common n-gram patterns. These analyses are focused on the trends, stylistic differences and distributions across genres and decades while also taking into consideration how these patterns change around part borders. They form the foundation for latent modeling and the different approaches in music information retrieval tasks presented in 8.4 and 9.

8.3.1 Structural Features

We begin with an overview of the dataset's quantitative characteristics and metadata coverage, shown in Table 8.1, in order to better understand its scope and composition. The dataset contains over 679,000 musical pieces collectively having close to 52 million chords. Its harmonic vocabulary consists of 749 chords and 3,577 inverted forms of those. Around, 397,000 of the included tracks are also accompanied by information about their structural segmentation (how they are split into parts), resulting in over 2.6 million individual parts. Moreover, considering additive annotations of metadata, the linkage with Spotify data is largely complete, having genre, release date, song and artist identifiers available for the majority of the tracks.

Table 8.1: Summary Statistics of the Dataset [14].

Description	Count
No. of Tracks	679,807
No. of Chords	51,994,634
No. of Unique Chords	749
No. of Unique Inverted Chords	3,577
No. of Tracks with Parts	397,580
No. of Parts	2,670,457
No. of Tracks with Release Date	422,181
No. of Tracks with Spotify Genres	429,753
No. of Tracks with Main Genre	352,111
No. of Tracks with Spotify Track ID	440,284
No. of Tracks with Spotify Artist ID	510,986

Following this overview, we examine the distributions of songs and artists across the twelve main genre categories. Histograms illustrating the numbers of unique songs, artists and the ratio between them are presented in Figure 8.3.1. The distributions of songs per genre is noticeably unbalanced, having genres such as Pop, Rock, Alternative, Country and Pop-Rock collectively accounting for the majority of songs assigned with a main genre. Similarly, the distribution of artists per genre is skewed, with the same genres of attributing more than two- thirds of all unique artists. It is important to note that these two distributions are not directly analogues. The songs-per-artist ratio reflects the relative representation of artists within each genre in the dataset. For instance, Pop-Rock artists contribute close to twenty songs on average, while Country and Rock artists average more than ten per artist. Most other genres average around six songs per artist, with Rap and Electronic exhibiting even lower values.

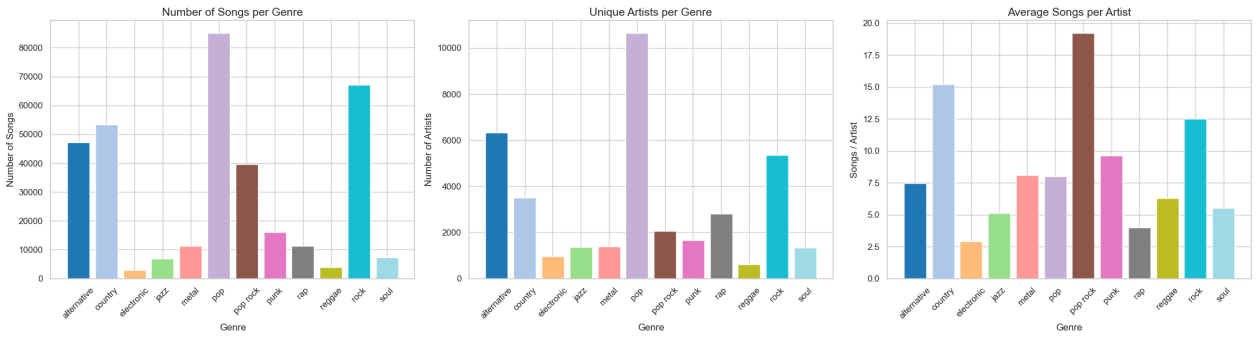
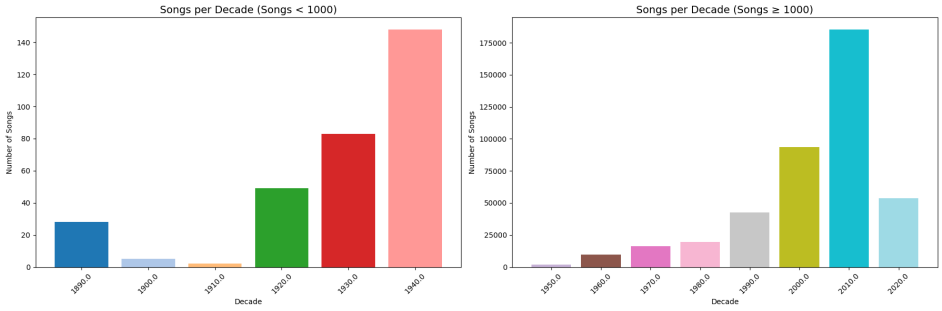


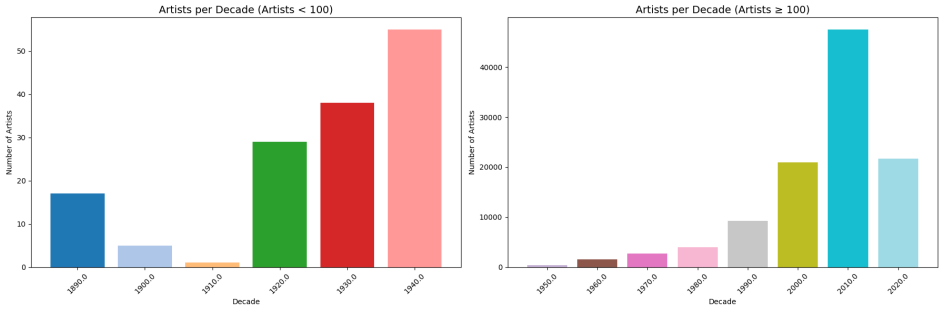
Figure 8.3.1: Song and Artist Distribution by Genre and Average Song number per Artist.

To complement the above statistics, Figure 8.3.2 shows the distribution of tracks and artists per decade. Each distribution is split into two parts because of the disproportional representation of songs and artists in the earlier decades compared to the more recent ones. The distribution of songs per decade has the first plot show decades with less than 1,000 songs, while the first plot for the distribution of artists per decade shows those decades with less than 100 artists representing them. The second plots, in both distributions, show the decades that surpass these thresholds. This separation allows both sparse and dense periods to be visualized clearly. Most songs and artists in the dataset are from the 2010s, followed by the 2000s and 2020s, collectively including the majority of the entries. The least represented decades are those prior to the 1950s, with the 1910s and 1900s being the most sparse. Unlike the genre-level distributions, the songs-per-decade and artists-per-decade distributions are largely proportional, with the 2020s being a notable exception due to fewer recorded tracks despite a moderate number of contributing artists.

Continuing the analysis about relationships between the sampled songs across genres and decades, in Figure 8.3.3, the songs that have both a main genre and a release date are used to present the distributions of each genre across decades. In agreement with the songs-per-decade distribution, the majority of songs in each genre belongs to the 2010s, with the 2000s being a close second in genres such as Country, Jazz, Metal,



(a) Song Distribution by Decade.



(b) Artist Distribution by Decade.

Figure 8.3.2: Distribution of songs and artists over the decades.

Pop-Rock, Punk, Reggae and Rock, while having less representation in the rest. Furthermore, most genres have little representation in decades prior the 1990s, with notable exceptions being Country, Jazz, Pop-Rock, Reggae and Rock where the representation accumulated in those decades is comparable with the rest.

Moving on from metadata about entire song entries, Figure 8.3.4 showcases distributions based on structural parts, sourced from entries of the dataset that include part information. The first plot shows the distribution of number of parts per song, resembling a normal distribution centered around 7, with the number of songs decreasing exponentially as parts increase, eventually becoming insignificant beyond 15 parts.

The second and third plots showcase the average number of parts in a song with respect to genres and decades. In both plots most of the averages span between 6 and 8 parts in a song, with Metal averaging a bit higher and Rap having the lowest average amongst the genres, while the 1890s average surpassing the 8 parts mark followed by the 1980s averaging around 7 parts per song.

Exploring one layer deeper, Figure 8.3.5 illustrates the distribution of part sizes measured in chords. Part sizes greater than 50 were excluded because of their negligible frequencies. In the first plot the overall distribution peaks around 8 chords per part, while also local peaks can be seen at lengths 4, 6, 12 and 16 and some smaller ones on 20 and 24. It is important to note that many of these local peaks occur in multiples of 4.

Part sizes across genres and decades can be observed in the second and third plots. The average part size spans between 12 and 14 for both plots. In the genres plot what is notable is the average in part size of Rap emerges higher than the rest even if the difference is negligible while previously Rap showed the smallest average number of parts per song. Considering the part size distribution across decades, the 1900s present an exception having an average higher than 16 chords per part while 1950 has the smallest value around 11 chords per part.

Following the previous plots, Figure 8.3.6 presents additional distributions about part types and the average lengths by type. In the first plot, where the distribution of part types is shown, it is evident that Verses and Choruses vastly outnumber the rest of the part types, while the second place is filled by Intros, Bridges and Outros followed by Instrumentals, Interludes and Solos having less than 100,000 appearances each. The

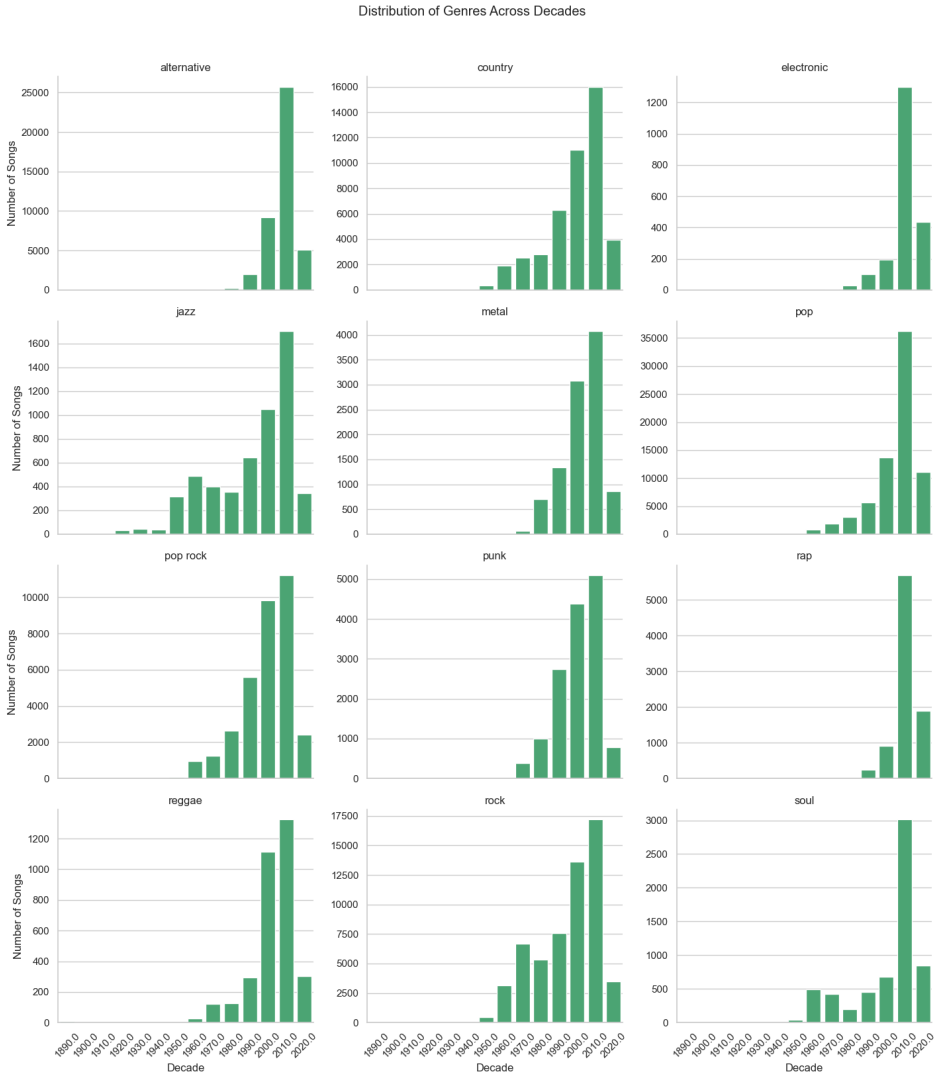


Figure 8.3.3: Distribution of genres across decades.

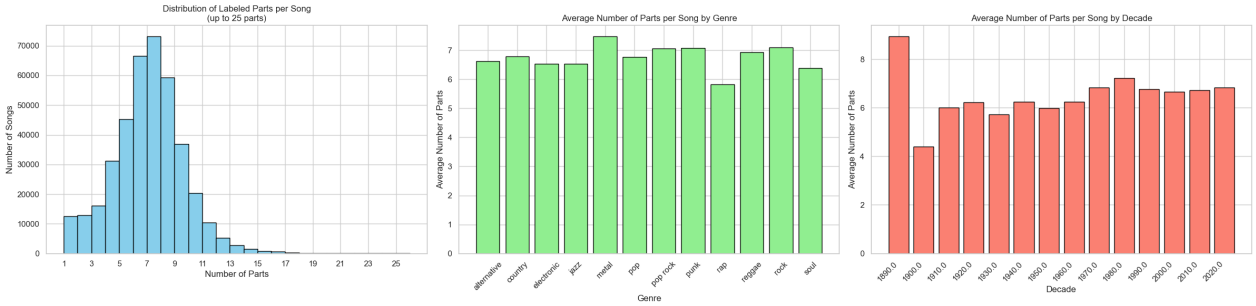


Figure 8.3.4: Number of parts per song, overall, per genre and per decade.

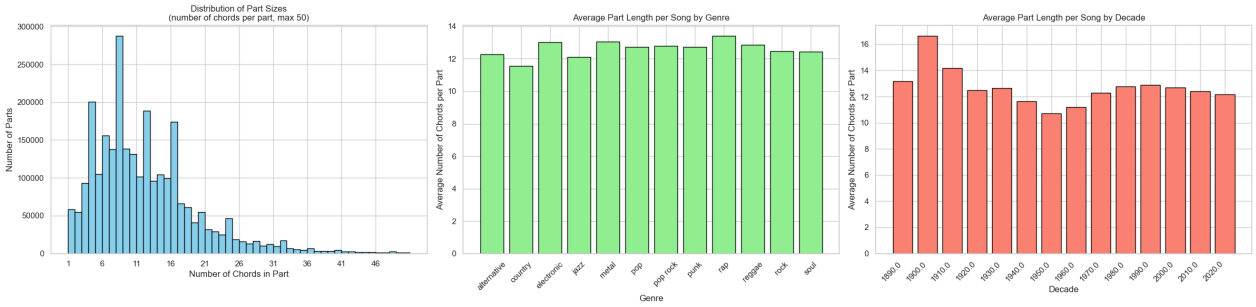
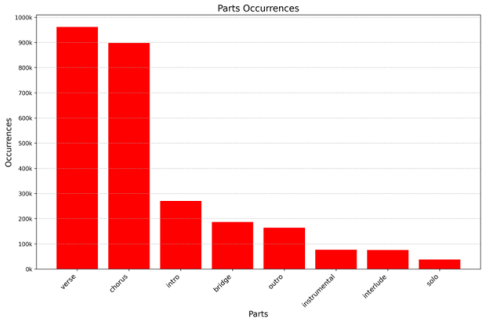
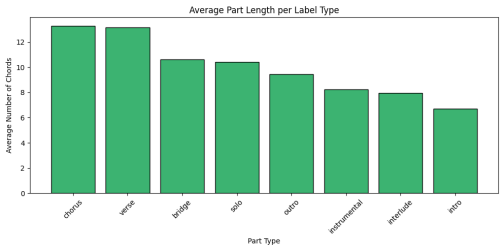


Figure 8.3.5: Part sizes overall, per genre and per decade.

second plot displays average part length per type. The averages span between 6 and 14 chords, with Choruses and verses averaging more than 12 and intros being the shortest averaging around 6 chords per occurrence. The two distributions seem to have the part types in the same order, with the exception of Intros that were moved to last.



(a) Distribution of part sizes (number of chords per part) [14].



(b) Average part length per label type.

Figure 8.3.6: Comparison of chord structure statistics: (a) part-size distribution and (b) average length by section type.

The next step in this analysis is to go into the chord level. In Figure 8.3.7 the distribution of unique chords per song is exhibited across genres and decades. Across genres it is noticeable that Jazz and Soul exhibit the highest averages of unique chords and standard deviation per song, while the rest are split into two groups: the ones with averages above 6 unique chords and standard deviation close to the top ones, them being Metal, Pop, Pop-Rock and Rock, and those with averages below 6 unique chords small standard deviation, meaning Alternative, Country, Electronic, Punk, Rap and Reggae. Across decades, both averages and standard deviations are close together spiking during 1920s and overall decreasing towards the 2020s with not that many notable changes. As to the decades prior to the 1920s, they show smaller values, both of the averages and the standard deviations, compared with the rest.

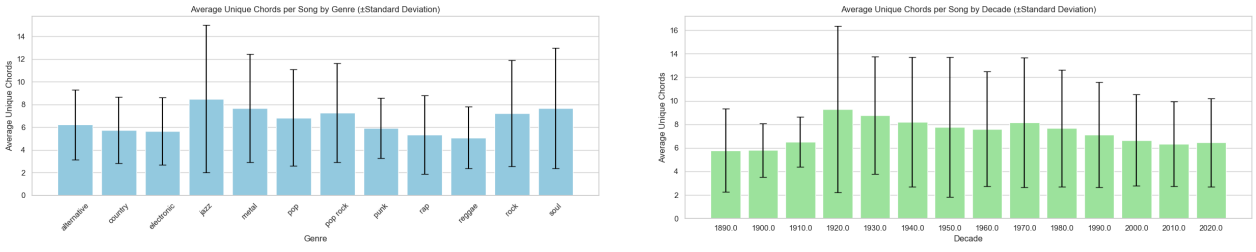


Figure 8.3.7: Average usage of unique chord per genre and decade.

Further chord information is shown in Figure 8.3.8, where the most frequent chords are presented alongside

their appearance percentages. It is evident that these chords, simple in their form, dominate the dataset, with no extensions or inversions using the most common qualities major (G, C, D, A, F, E) and minor (Amin, Emin, Bmin). Together, these top 9 choices account for two-thirds of the overall chord usage.

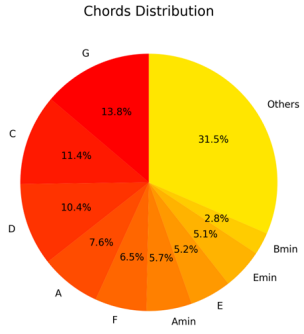
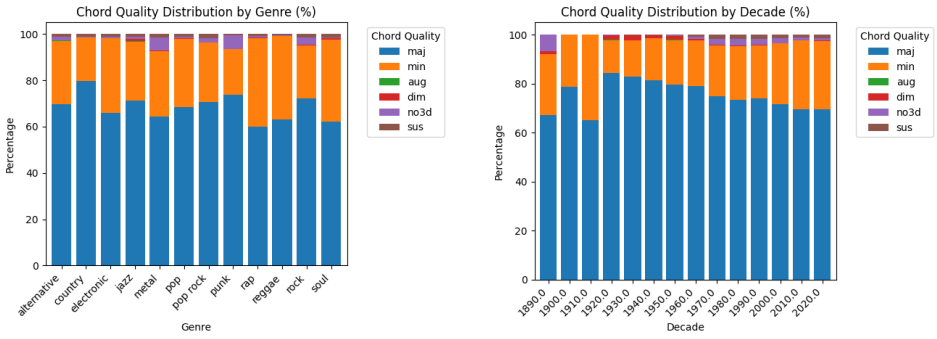
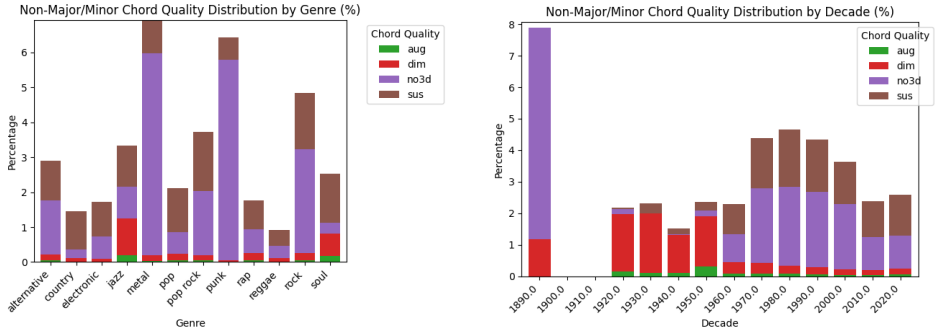


Figure 8.3.8: Distribution of chord usage [14].

Moving forward, the analyses focuses on the components of the chords, them being the qualities, extensions and the inversions. Starting with the qualities, their distribution is presented in Figure 8.3.9. The top part of the figure shows the distribution of qualities over the different genres and decades. It is clear that the majority consists of major chords, with minor chords being the second most prominent quality type and together they constitute over 90% across all categories. The ratio of Major and Minor types changes a small amount across genres, while it seems that minor chords are covering more space moving from 1920s to 2020s.



(a)



(b)

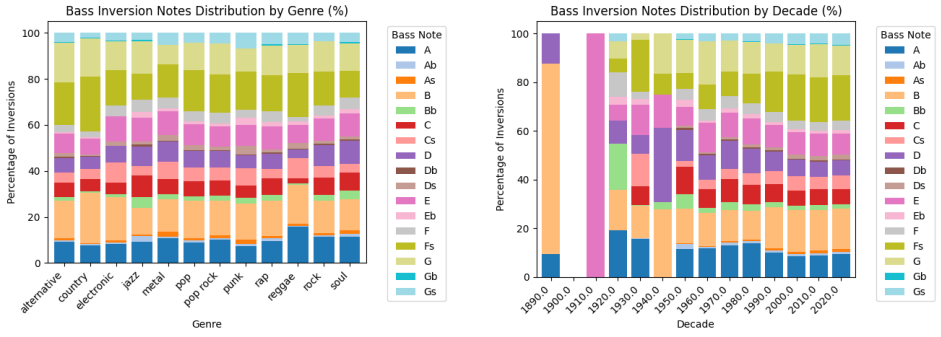
Figure 8.3.9: Distribution of (a) all qualities and (b) outlier qialities, across genres and decades.

To investigate the outliers, their distributions are shown in the bottom part of Figure 8.3.9. As can be seen in these plots, in most genres these irregular qualities do not even exceed 5%, exceptions are Metal, Punk and Rock, where power chords are the most dominant. Power chords seem to dominate in half the genres, while the other half either feature more Suspensions or show an equal number of both. The Augmented and

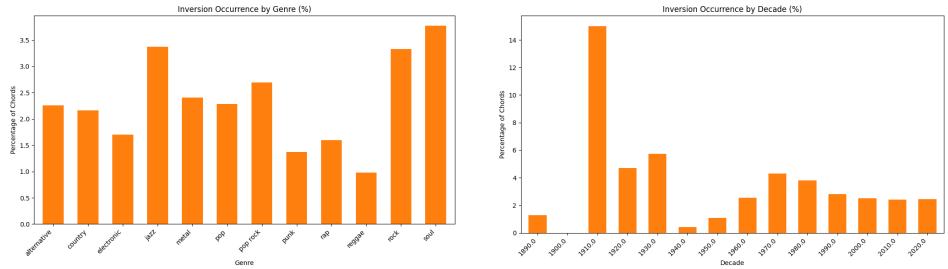
Diminished qualities appear in an insignificant amount besides the genres of Jazz and Soul, where Diminished chords have comparable proportions to the Power and Suspended chords. As for the distributions across the decades, Diminished chords seem to be more popular prior to the 1960s, with Power and Suspended ones taking charge after that period.

Subsequently, we consider the inversion occurrences in the dataset. As presented in the top part of Figure 8.3.10, the distribution of inversion choices remains consistent across genres and decades, with $B, F\#$ and G being picked slightly more as the bass notes. Some inconsistencies appear in the distributions over the decades prior to the 1920s, showing complete dominance of specific notes on bass, such as E during the 1910s, B during the 1890s and a gap in the 1900s.

The bottom part of Figure 8.3.10, shows the percentages of inversion usage over the categories. Across genres the percentages do not seem far off with the top ones being Jazz, Rock and Soul around 3.5%, while the lowest percentages correspond to Punk and Reggae roughly at 1%. Over the decades it is evident that a local peak appears around the 1970s, decreasing slightly towards recent times, while percentages in earlier decades also appear low, besides the 1910s where they skyrocket to 14% and the 1920s and 1930s that have similar inversion representations as the 1970s.



(a)



(b)

Figure 8.3.10: Distribution of (a) inversion choices and (b) inversion existence percentages, across genres and decades.

Finally, extensions distributions are displayed in Figure 8.3.11. In both plots it can be seen that the most common extension is adding the 7th note to the chord, with a considerable gap to adding the 9th, that falls into second place for the majority of categories, while it also seems that 13th is a more popular choice than 11th. The most notable observations are that Jazz and Soul seem to utilize extensions more than the rest of the genres, while Metal and Punk use them the least. Considering the decades, extension usage seems to peak around the 1920s, having lower percentages prior, and gradually decrease as we approach recent times.

8.3.2 Sequential Features

In this section, we present the statistics that resulted from the analysis focused on local dependencies in chord progressions, meaning bigrams and trigrams, ngrams of higher order, specifically 4,5 and 6-grams, are used in 8.4.

Firstly, in Figure 8.3.12 we present the distribution of root note motions between chords transitions. In

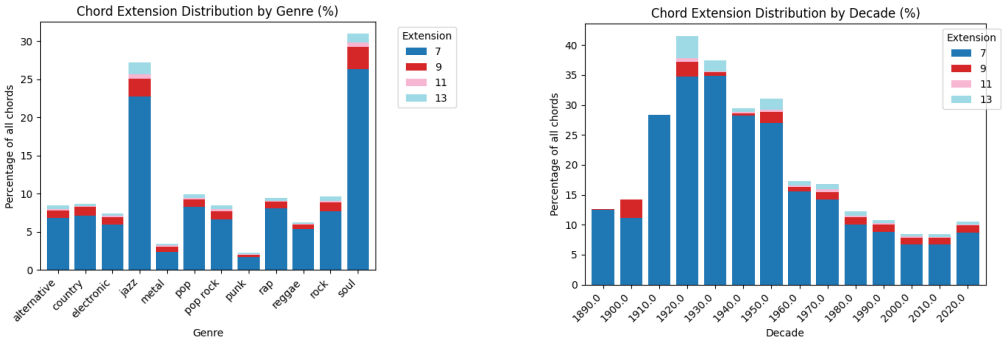


Figure 8.3.11: Distributions of extensions across genres and decades.

both plots, one describing motions across genres and the second one across decades, it can be observed that there is a preference in motions of 5 semitones, followed by motions by 2 semitones, with the first almost being twice the percentage of the second. Motions of 3 and 4 semitones seem to be equally picked while the rest reside lower, with motions of 6 semitones being the most rare and almost being infinitesimal. Another notable observation is that the preference between moving up or down in semitones appears to be minimal across categories with the highest difference being around 2% for the motion of 2 semitones.

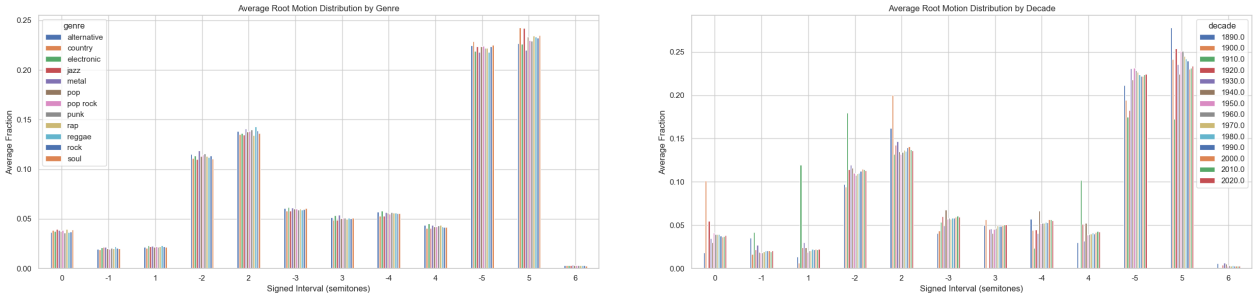


Figure 8.3.12: Root motions measured by semitones across genres and decades.

Proceeding with sequential analysis, we collected the most common bigrams and trigrams focusing on tracks with valid part labels. The top 20 most common ones are displayed in Figure 8.3.13, where for each n-gram two columns are presented, the first has the most common n-grams overall and the second the most common ones between part transitions. Those between transitions have the last chord of the n-gram belonging to the next part of the song as a way to measure how the context changes between parts. It is evident in both tables that the distribution of the top frequencies loses its structure when compared to the frequencies across parts. The most common bigram *C*-*G* moves from first place to third while other bigrams are reaching the top spots such as *E*-*A* moving from being 13th to 7th place. This change in distributions can be clearly seen in trigrams, where the first place (*G*-*C*-*G*)is taken over by the 11th (*D*-*G*-*C*) one when switching from overall to across parts, with the previous last place becoming 4th, while a good amount of the top 20 trigrams overall do not make the top 20 between part boundaries.

To investigate even further, in Figures 8.3.14 and 8.3.15 we present the 10 most common chords and bigrams and the distributions of the top 5 next chords selected in this sequence both within each part and across parts. The across parts distributions placed the chord or bigram in question as the last elements of a part and checking which chord followed them as the first of the next part.

By computing the Total Variation Distance (TVD) 8.3.1, we can see that the within and across distributions differ for bigrams by approximately 52% (Overall) and 17% (frequency-weighted) and for trigrams by approximately 57% and 23%. The formula of the total variation distance between two distributions can be seen below:

Top 20 Chord Bigrams — Overall vs Across Parts

Rank	Overall (Bigram)	Count	Across Parts (Bigram)	Count
1	C → G	1223564	G → C	122346
2	G → D	1014246	D → G	103336
3	G → C	980146	C → G	70224
4	D → G	852062	A → D	68636
5	F → C	729076	G → D	63256
6	D → A	637851	C → F	60017
7	A → D	565292	E → A	54259
8	C → F	521201	F → C	45411
9	A → E	483153	D → A	41526
10	G → Amin	468978	G → Amin	40044
11	C → D	447004	A → E	35774
12	F → G	434500	D → Emin	33164
13	E → A	424090	D → C	30465
14	Emin → C	355760	G → F	30377
15	D → Emin	352211	G → Emin	28707
16	Amin → G	379815	C → Amin	28310
17	D → C	377039	B → E	26214
18	Amin → F	376451	C → Emin	21673
19	G → F	365110	A → G	21601
20	C → Amin	346017	Amin → C	20189

Top 20 Chord Trigrams — Overall vs Across Parts

Rank	Overall (Trigram)	Count	Across Parts (Trigram)	Count
1	G → C → G	354285	D → G → C	32600
2	C → G → D	306824	C → G → C	28711
3	C → G → C	298049	G → C → F	24613
4	D → G → D	274881	G → C → G	24430
5	F → C → G	263670	C → D → G	23931
6	G → D → G	250124	D → G → D	23594
7	C → F → C	216719	F → G → C	23482
8	C → G → Amin	209126	A → D → G	22613
9	G → D → A	196786	G → D → G	22461
10	A → D → A	190976	G → A → D	17267
11	D → G → C	190470	A → D → A	16085
12	C → D → G	190050	D → A → D	14861
13	G → D → Emin	182484	E → A → D	14668
14	F → G → C	178555	A → E → A	14590
15	Emin → C → G	176381	C → G → Amin	14506
16	D → A → D	173577	F → C → F	13780
17	Amin → F → C	169207	E → A → E	13068
18	D → C → G	164655	G → D → Emin	12484
19	G → C → F	160997	Amin → F → C	11859
20	F → C → F	159881	C → F → C	11566

Figure 8.3.13: Top bigrams (left) and trigrams (right) over all and on part transitions.

$$\mathrm{TVD}(P, Q) = \frac{1}{2} \sum_{x \in \mathcal{X}} \left| P(x) - Q(x) \right| \tag{8.3.1}$$

where P and Q are probability distributions over the same set of events $\mathcal{X}$, and x represents each possible event.

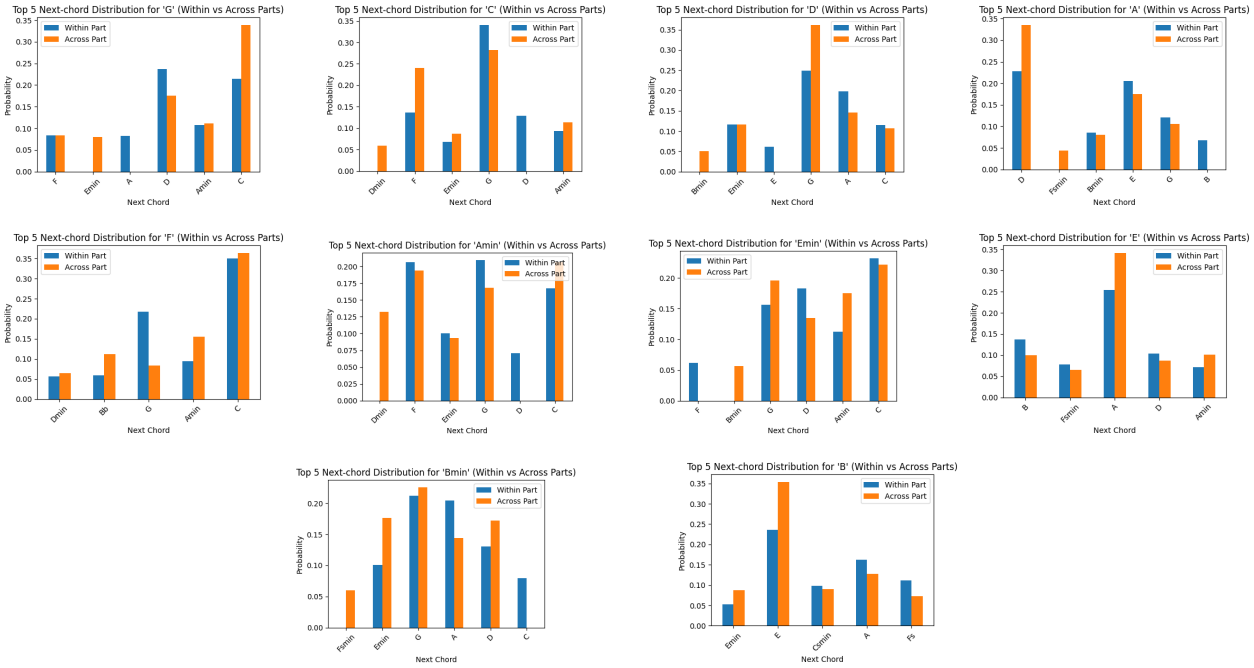


Figure 8.3.14: Top 10 chords and how the next chord distributions change on part transition.

With regard to bigrams, remarkable case constitutes the G chord having its most common successors within parts, the D chord with approximately 24% with the C being close second with 22%, swap places and the C chord dominating with 34% over the 17% of the D chord. Additionally, following the *Amin* chord from the within distribution we can see that F and G chords are the top candidates with around 20%, with the C chord placing third having 17%. This dynamic changes across parts with the C chord taking the first place swapping with the G chord, while the F chord's percentage drops below 19%. Other interesting observations are that in most cases, such as the D, A, E and B chord, where the most prominent choice within parts gets empowered across parts. In the rest instances the dominant choices remain the same, while the distribution hierarchy of the less influential ones varies. For example after the F chord, the C chord remains the most common option but the G chord moves from second to fourth most common losing roughly 16%, similar to

the A and D chords losing approximately 6% and becoming the fourth choice, instead of second, after the $Bmin$ and $Emin$ chords, respectively. Considering the C chord, it seems that the distribution across parts closes the gap the occurrences of F and G chords to 5%, from the 20% they had within the same part.

Continuing with the trigram patterns shown in Figure 8.3.15 we detect greater variations between the within and across parts distributions. The most notable of them are the changes after the progression C-G where the most prevalent choice changes from being D and C with 27% and 24% respectively in the within distribution to C with 34% and D with 13% in the across distribution. In a similar manner the most prominent chords after the progressions G-C, D-G, F-C and A-D change from being G, D, G and A to being F, C, F and G, by growing by 16%, 16%, 11% and 16% respectively. The rest trigrams show little variation of the percentages of the dominant candidate chords, while the others adjust insignificantly.

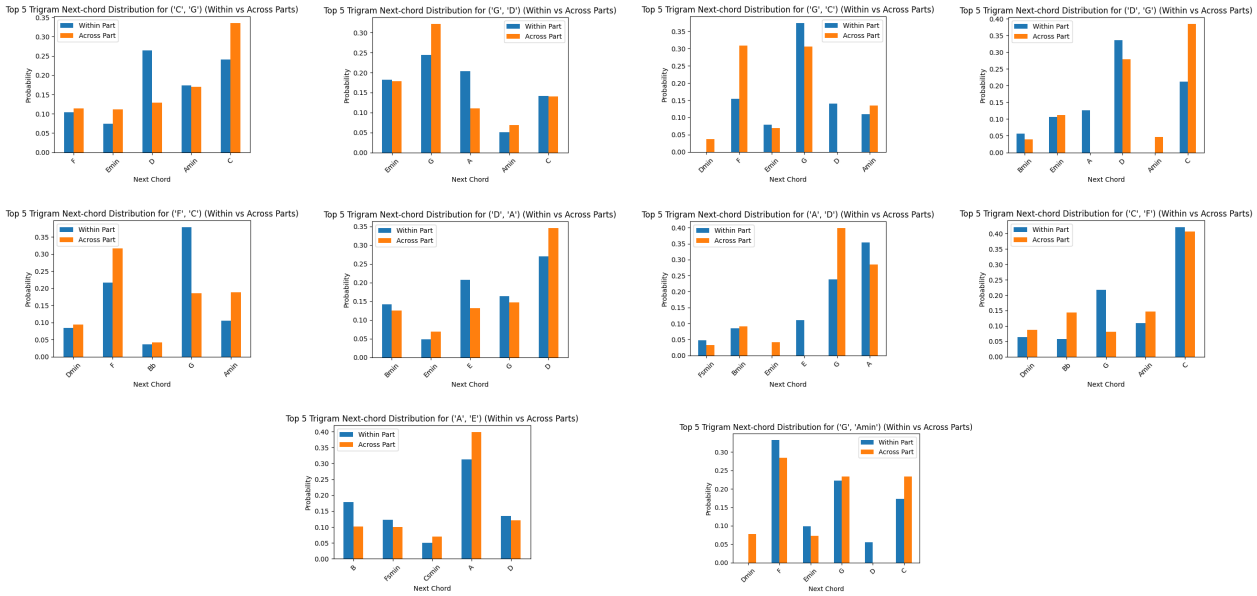


Figure 8.3.15: Top 10 bigrams and how the next chord distributions change on part transition.

8.4 LDM analysis

This section describes the preprocessing of the dataset and the methods applied to create a graph-like structure from n-grams to model songs using latent distance models with respect to their genre and decade of release. The intuition here is that we can design a method to represent each song in a latent space through its n-grams and try to see if this representation aids classification tasks such as genre and decade classification with the results being shown in Chapter 9.

8.4.1 Dataset Balancing

To ensure that each category was equally represented, the first step was to clean and balance the raw dataset. Entries that had neither genre or decade metadata were discarded alongside those with invalid chord symbols. The remaining data was then balanced by category, retaining the same number of samples per valid group. Groups with few samples, less than 200, were also excluded to avoid having underrepresented categories. The final dataset consisted of 2,814 samples per genre category and 1,748 samples per decade category. Subsequently, stratified train/test splits were performed using 80% of the samples for training and 20% for testing, to preserve the category distributions in both splits.

8.4.2 Chord Processing and N-gram Extraction

Before encoding each chord, they were preprocessed to standardize their representation. Structural part labels were discarded and only the root of each chord was kept, removing quality, extension and bass information.

Each root was then converted to an atonal pitch class, using integers from 0-11 to represent each chord, capturing the twelve semitones in an octave. The atonal representations help mitigate the combinatorial complexity that follows n-grams as the n increases. After simplifying each sequence, overlapping n-grams were harvested from it for $n = 4, 5$ and 6, focusing on capturing local patterns. For each n-gram its absolute frequency across all songs and relative to each category was measured.

8.4.3 Category Graph

Before constructing the full n-ram graph, a smaller graph was created where each node represented a category and edges reflected the relationships between them. The scope of this graph was for direct initialization of the latent parameters of the full n-gram graph instead of a random one, each n-gram starts closer to it's most relevant category. The edges of this smaller graph were defined soft Jaccard similarity of their n-gram profiles, where the numerator summed the minimum percentage contributions of shared n-grams and the denominator summed the maximum contribution across all n-grams. This metric captures both the presence and relative importance of shared patterns across categories. Finally, edges were normalized to a fixed scale, resulting in a weighted, undirected graph encoding category similarities.

8.4.4 N-gram Graph

For the n-gram graph, each node represents a unique n-gram and edges were established based on category-specific nearest neighbors. In more detail, within each category, n-grams were sorted by normalized percentage contribution and total count, and pairs of n-gram with neighboring ranks (rank differences ≤ 25) were connected. A separate edge set was created connecting each n-gram with the category it represents the most, producing another weighted undirected graph that will later allow category nodes to take their place in the latent space amongst the n-grams.

8.4.5 Node Coloring and mapping

To enhance interpretability, each n-gram node was assigned a categorical color corresponding to the category in which it had the highest relative contribution. This information is extracted from the connections between n-grams and categories described above. This coloring provides a way to visually identify clustering inside the latent space and highlight n-grams that are strongly associated with specific categories.

Additionally, each n-gram node was assigned a mapping metric to quantify its distribution across categories. A Kullback-Leibler divergence based measure 8.4.2 was computed from the n-gram's percentage contributions, capturing its relative specificity to particular categories. These mappings enable new unseen songs to connect with the n-gram graph and be positioned in the latent space and interpreted in terms of its similarity to the existing dataset. The Kullback-Leibler divergence-metric formula is a follows:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{c=1}^{C} p_c \log \frac{p_c}{q_c}, \tag{8.4.1}$$

where p_c is the observed percentage of an n-gram in category c and q_c is a reference distribution.

In our implementation, we use a uniform reference distribution $q_c = \frac{1}{C}$, which simplifies the formula to:

$$\text{KL-inspired}(x) = \sum_{c=1}^{C} p_c \log p_c + \log C. \tag{8.4.2}$$

This formulation measures how concentrated an n-gram is across categories:

- Values close to 0 indicate that the n-gram is evenly distributed across all categories.
- Higher values indicate that the n-gram is more category-specific, contributing predominantly to a subset of categories.

Using this metric we assess how much information each n-gram contributes to the latent space and how strongly its existence in a song influences how it should be categorized.

8.4.6 Latent Distance Modeling

After determining the category, n-gram and n-grams to categories relationships, the latent distance model was trained to embed both categories and n-grams into a shared latent space. The training process was structured in three sequential phases to produce interpretable and semantically meaningful embeddings. The models were implemented in Python and trained using PyTorch with the Adam optimizer. training lasted for 100 epochs with 100 steps per epoch, using a learning rate of 0.01 and batch sizes of 4,000 for n-grams and the number of categories for category graphs. It is important to note that in each step of each epoch, the model only trained the positions of the selected n-grams, determined by batch size, relative to each other. The implementation is based on the latent distance model available at github [15], that uses a custom negative log-likelihood loss, derived from pairwise distances in the latent space.

Phase 1: Training the Categories Graph

The first phase is focused only on category nodes. Training the categories graph positioned each category node in the latent space was placed closer to the category nodes that were most related, as shown in Figure 8.4.1.

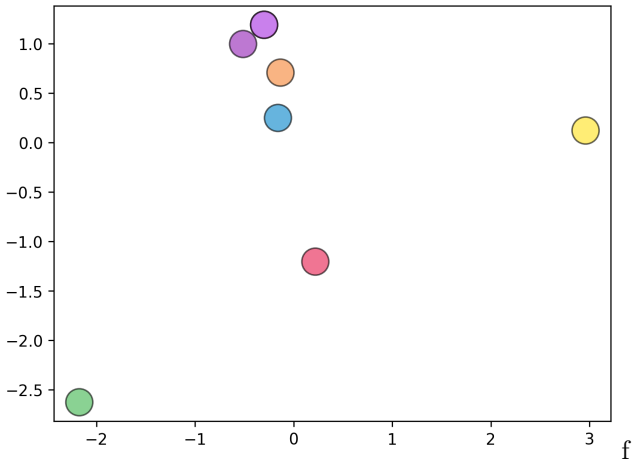


Figure 8.4.1: An example of category nodes after the first training phase.

Phase 2: Training the N-grams Graph

The second phase leverages the latent positions of categories computed in the first phase to initialize the latent positions of the n-grams. Each node is initialized at the position of the category node it is most associated with while also adding some small noise. Training the n-gram graph results in their final positions in the latent space, as shown in Figure 8.4.2.

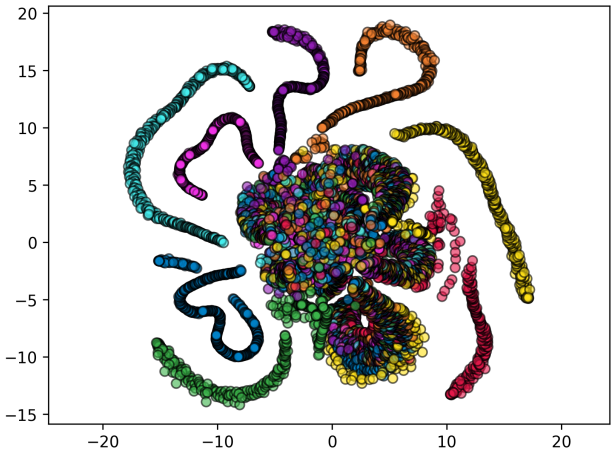


Figure 8.4.2: An example of n-gram nodes after the second training phase.

Phase 3: Repositioning the Categories

After the n-grams have reached their optimized positions in the latent space, the category nodes are repositioned based on their connections to the n-grams. As mentioned previously each n-gram is linked to a single category for coloring, allowing the final position of each category to be computed as a weighted centroid of its connected n-grams, as shown in Figure 8.4.3.

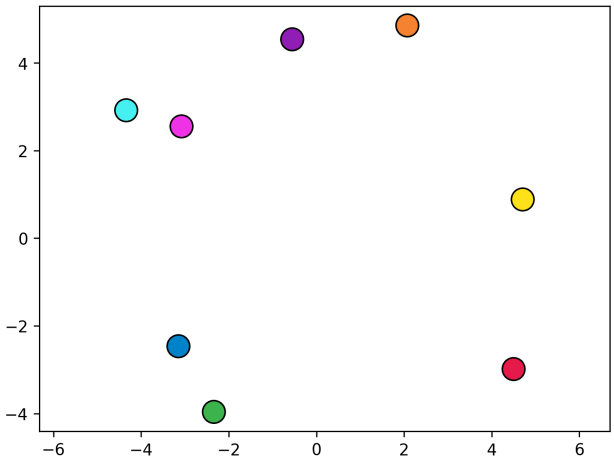


Figure 8.4.3: An example of category nodes after the third training phase.

The result of the above phases is a latent space where n-grams and categories occupy optimized positions, revealing interpretable geometric relationships that reflect their statistical and semantic associations. Below, in Figures 8.4.4 and 8.4.5, the latent positions of 4,5 and 6-grams can be observed as well as the latent positions of the genre and decade categories respectively.

From the plots it evident that stronger, more visible clusters are forming while we move from 4-grams to 6-grams. The dense regions closer to the center account for roughly 90% of the total points. In the genres case we have a total of 11,117 unique 4-grams, 29,730 unique 5-grams and 45,462 unique 6-grams. Considering the decades we count 7,490 unique 4-grams, 16,436 unique 5-grams and 21289 unique 6-grams.

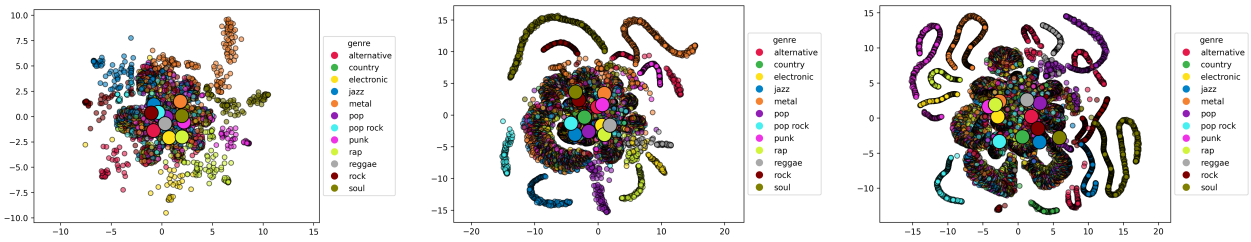


Figure 8.4.4: The latent positions of 4-grams, 5-grams and 6-grams, from left to right, based on connections created by genre similarity.

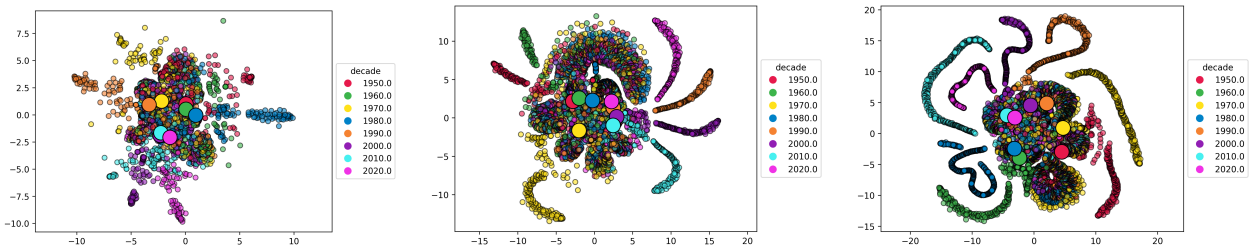


Figure 8.4.5: The latent positions of 4-grams, 5-grams and 6-grams, from left to right, based on connections created by decade similarity.

Focusing on the 5-gram and 6-gram plots where we can see better differentiation of the categories some patterns can be observed. The most notable ones are the Punk and Metal genres being positioned close in both occasion, while the same can be said for Jazz and Country. Additionally, Rock and Soul seem to have similar n-gram preferences as well as Reggae and Pop. As for the decades, we can clearly see 1960s and 1980s being close and that 2000s 2010s and 2020s are adjacent to each other in both 5-gram and 6-gram plots, while the 1990s reside in proximity. The 1970s seem to distance themselves from the formed groups in both cases.

Chapter 9

Experiments

In this chapter we present the experiments conducted to evaluate our proposed framework. The experiments are divided into two main tasks: classification and chord prediction. All implementations were in Python, using PyTorch library for model construction and training. For optimization, we employed the Adam optimizer, which provides adaptive learning rates and efficient convergence for both the recurrent models and the latent distance models. The latent distance model used a negative log-likelihood loss function, as described in 8.4 while the recurrent models, used cross-entropy loss, described in 4.2.

9.1 Genre and Decade Classification

In this section, we mainly describe the pipeline for the sequential models training as the latent distance modeling pipeline for preprocessing and training was described in 8.4. In the sequential models we include the recurrent models and the state-space model architecture of Mamba. The evaluation methods are described for both architectures.

9.1.1 Data Representation and Preprocessing

For the sequential models, similar measures to the one used for the latent space models, towards preprocessing the raw data and preparing them for training, were employed. The dataset went through a balancing and cleaning stage for both category classification tasks, and each chord was turned into an integer corresponding to the root's pitch class 0-11. In contrast to the latent distance model procedure, the dataset was not split into n-grams and whole song sequences were use for training.

9.1.2 Model Architectures

Four types of sequential architectures were considered, the vanilla neural networks, the gated recurrent unit networks, the long short-term memory networks and the Mamba model. Each model received sequences of chord integers (0-11) as input processed them through an embedding layer to learn meaningful representations of each chord and produced a probability distribution over categories through a fully connected output layer with softmax activation for categorical classification. The main module consists of one layer of the main computation sequential units.

9.1.3 Training

Models were trained using the cross-entropy loss function and hyperparameters were tuned empirically for optimal performance. The embedding dimension used was 4 and the hidden dimension of the model was 256, training lasted up to 100 epochs, using batch size of 256, with early stopping and learning rate scheduling starting from 0.001 for smoother training. Mamba models include some extra hyperparameters, we used state vector of size 32, convolution kernel size of 4 and expansion of the model of size 2.

The dataset was split into training, validation, and test sets (80/10/10). The validation set was used to monitor the model's performance during training, with the model state achieving the lowest validation loss being saved and later used for evaluation on the test set.

9.1.4 Evaluation

For both latent distance and sequential models, classification performance was assessed using accuracy and confusion analysis on the test set, identifying common misclassifications between categories.

For the recurrent evaluation, test loss was also recorded to compare model performance. For the latent distance models the predicted class for each song was obtained by calculating distances in the latent space from the category nodes positions learned during training and choosing the minimum. For the recurrent methods the prediction was obtained from the maximum probability output by the softmax layer.

9.2 Chord Prediction

The chord prediction task relies on symbolic chord sequences from the Chordonomicon dataset. The experiments were designed to assess how training set size and sequence length influence model performance. Each song in the dataset is represented by a progression of chords accompanied by structural section labels when such annotations are available (<Verse_1>, <Chorus_2>). To prepare the data for sequence modeling, their raw form had to be preprocessed in order to produce the two different representations we used to experiment.

9.2.1 Data Representation and Preprocessing

The raw dataset was filtered to remove incomplete or noisy entries. Songs with at least genre or decade metadata were preferred while ones containing odd chord symbols (having indications of inversion without stating the bass note "/ ") were discarded.

Normalization and Cleaning

The first step was to convert part labels such as <Verse_1>, <Chorus_2> to canonical forms Verse, Chorus, by removing the brackets and the numeric suffixes. These labels were retained only for the creation of section-wise chord progressions and were excluded from the main chord vocabulary.

Vocabulary Construction and Encoding

In the experiments we tested two different representations of the chords, one treating each unique chord as a token and the other decomposing the chord into three parts Root, Quality + Extensions and Bass.

- **Single-token representation:** Each unique chord symbol (e.g., `Cmaj7`, `Am`, `G/B`) was treated as an atomic token. This approach captures complete harmonic information within a single unit, resulting in a compact sequence representation.
- **Three-token representation:** Each chord was decomposed into three components:
  - **Root:** the pitch class of the chord (e.g., C, D#, F#);
  - **Quality + Extensions:** the chord type and added tones (e.g., maj7, min, sus4, add13);
  - **Bass:** the bass note when specified (e.g., the E in `C/E`), or `N` for none.

  Each component was encoded as an integer identifier, and mappings between component identifiers and complete chord identifiers were stored for decoding.

Dataset Construction

After filtering out songs without metadata, three datasets were generated sequentially:

- **Filtered dataset:** Contains all songs with at least one metadata field (genre or decade), with structural labels removed; suitable for evaluation on entire songs.

- **Labeled-song dataset:** Subset of the filtered dataset containing only songs with structural labels. After that the labeled-songs are used to create the segmented dataset, and are later kept without labels for evaluation.
- **Segmented dataset:** Each entry corresponds to a single labeled section extracted from the labeled-song dataset (e.g., a single verse or chorus). Labels and associated chord sequences are preserved for section-level modeling and evaluation.

All models were trained on the segmented dataset, while evaluation was performed on the filtered dataset, the labeled-song dataset, and the segmented dataset after filtering songs that had at least one part belong to the training or validation dataset.

9.2.2 Model Architectures

For the tasks four architectures were considered, the vanilla recurrent neural networks, the gated recurrent units, the long short-term memory networks and Mamba. The models receive sequences of either encoded chords (1-token) or encoded chord components (3-tokens) and uses an embeddings layer to create meaningful embeddings to be processed by the main module. The module's output is then collected by fully connected layers, with either one head per chord or 3 heads, one per chord component. For the 3-tokens representation the heads of the components are then combined to determine which chord from the known vocabulary fits best. The multi-head approach allows the model to leverage both component-level and full-chord supervision simultaneously.

9.2.3 Training

Models were trained using cross-entropy loss, with the total loss of the 3-tokens approach being the summation of the component and full chord losses. Hyperparameters were as follows: hidden dimension 256, one main module layer, batch size of 4096, learning rate 0.005, a maximum of 200 epochs and embedding dimension of 128 for the chords and 16 for each component.

To improve performance and generalization, added mechanisms were employed. Early stopping in combination with learning rate scheduling was used to avoid overfitting, while a stratified splitting ensured training validation and test sets had no overlapping songs and maintained label distributions. Also, during all phases, to increase robustness to variable-length sequences, sequences were cut randomly when sampling while ensuring the target chord followed at least 4 chords.

9.2.4 Evaluation

For both experiments the average accuracy and loss was used to measure performance over multiple runs. To enrich the evaluation, confusion analysis was conducted on mismatches to measure their influence in the model's performance.

9.3 Results

In this section we present the results of our experiments. The first part focuses on the classification task, while the second part addresses the chord prediction task.

9.3.1 Genre and Decade Classification

Table 9.1 shows the accuracy and loss of the trained models for both genre and decade classification tasks. The results can be grouped into two categories. In the first group, the latent distance model and the vanilla recurrent neural network achieve similar results, barely surpassing random guessing being 0.125 for the eight decades and 0.08 for the twelve genres. The second group, comprising of the gated recurrent unit, the long short-term memory network and Mamba architectures, achieves significantly better performance, nearly doubling the latent distance model's accuracy in the genre classification task and improving decade classification accuracy by around 10%. Notably, the gated recurrent unit exhibits lower loss values than the other models, even if it is not the top performer in terms of accuracy in the decade experiment.

Table 9.1: Accuracy and loss for each model grouped by category.

Category	Model	Accuracy	Loss
Genre	LDM	0.0917	-
	RNN	0.1190	2.4610
	LSTM	0.1670	2.3915
	GRU	**0.1889**	**2.3608**
	Mamba	0.1753*	2.3905*
Decade	LDM	0.1326	-
	RNN	0.1344	2.0814
	LSTM	**0.2259**	1.9623*
	GRU	0.2252*	**1.9557**
	Mamba	0.2223	1.9831

Best results are in **bold**, second best are marked with *.

To further examine model performance, we present confusion heatmaps in Figure 9.3.1. The heatmaps were generated by combining the evaluation mismatches from all models with the combined matrix showing over 90% agreement with each individual model's confusion matrix.

In the genre heatmap, certain pairs of genres show higher similarity than others. Jazz is most often confused with Country, and also showing multiple confusions with Soul. Additionally, Punk is frequently confused with Metal. Overall, most of the models' errors involve misclassifying other genres with Country.

In the decades heatmap, the 1960s show strong similarity with 1950s, with the 1950s being the most common misclassification. Another small cluster can be seen observed among the 2000s, 2010s and 2020s.

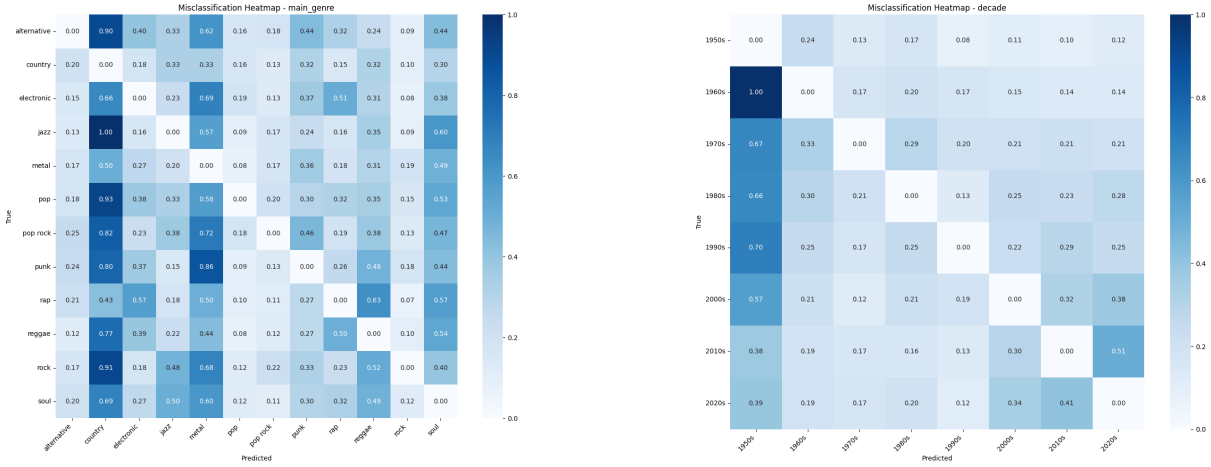
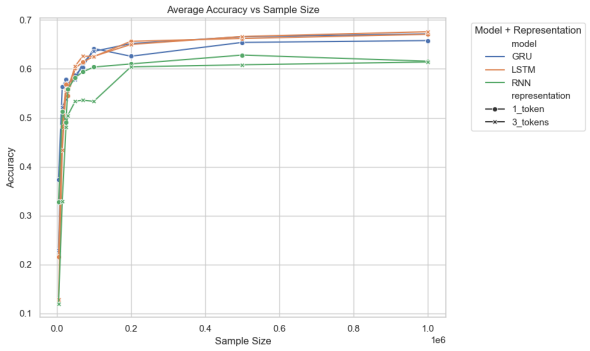


Figure 9.3.1: Confusion heatmaps for the genre classification task (left) and the decade classification task (right).

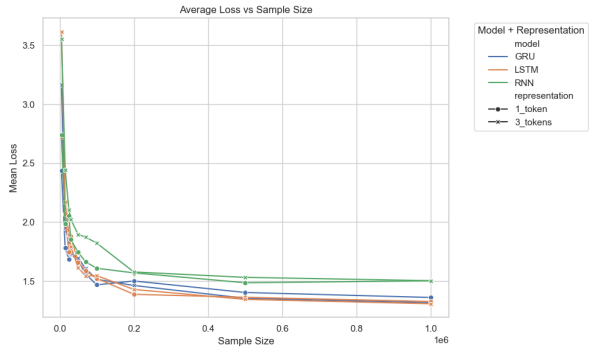
9.3.2 Chord Prediction

The chord prediction experiments consist if two main analyses. First, we investigated how the size of the training dataset influences model performance, as shown in Figure 9.3.2. Performance improves dramatically as the sample size grown from a few thousand entries (specifically 5,000) to 200,000, with only minor gains beyond that up to 1,000,000 samples. Vanilla recurrent neural networks consistently perform worse than the other models, while the 3-tokens representation has a slight outperforms the 1-token representation.

Table 9.2 summarizes the average results over multiple runs using the best performing sample size (1,000,000 samples). When tested with whole song sequences that have part labels, accuracy decreases and loss increases.



Cumulative Accuracy



Cumulative Loss

Figure 9.3.2: Cumulative Accuracy and Loss for the Standard dataset.

Performance drops further when tested on datasets missing structural labels. The 3-tokens representation generally outperforms the 1-token representation, except for the vanilla recurrent neural models. In terms of accuracy, the long short-term model achieves the best results, while the gated recurrent unit networks shows slightly better performance in loss despite a minor gap in accuracy.

Table 9.2: Average Accuracy and Loss per RNN Model and Representation across Datasets.

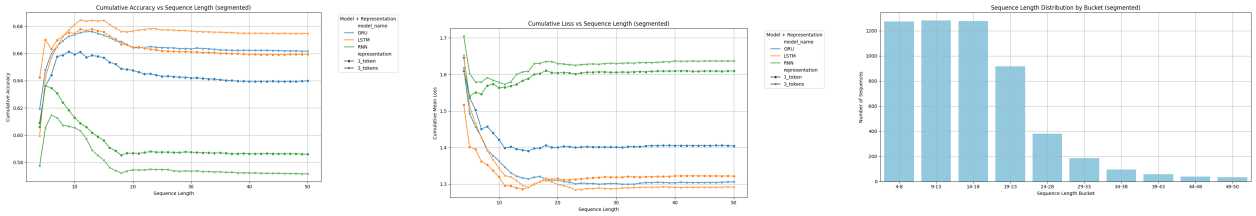
		Segmented		Labeled		Filtered	
Model	**Representation**	**Avg Acc**	**Avg Loss**	**Avg Acc**	**Avg Loss**	**Avg Acc**	**Avg Loss**
RNN	**1-token**	0.5859	1.63698	0.5632	1.6081	0.5448	1.6655
	3-tokens	0.5715	1.6370	0.5481	1.6504	0.5281	1.6972
LSTM	**1-token**	0.6594	1.3221	0.6244	1.3319	0.6007	1.4037
	3-tokens	**0.6748**	**1.2919**	**0.6394**	1.3079*	**0.6121**	1.3818*
GRU	**1-token**	0.6399	1.4046	0.6017	1.3832	0.5791	1.4479
	3-tokens	0.6618*	1.3060*	0.6311*	**1.2961**	0.6055*	**1.3782**
Mamba	**1-token**	0.5995	1.5165	-	-	-	-
	3-tokens	0.6501	1.4054	-	-	-	-

Best results are in **bold**, second best are marked with *.

Figure 9.3.3 illustrates how input sequence length affects prediction performance using average cumulative accuracy and loss. Average cumulative accuracy and loss, reflect how the model benefits from having more chords available before needing to make a prediction. The evaluation in this experiment was conducted on the segmented dataset. Very short sequences, such as those length of 4, provide insufficient context, leading to lower accuracy, while moderately longer sequences give the model a richer context and significantly better predictions. Vanilla recurrent neural networks tend to degrade beyond sequence lengths of about 7 chords, whereas other models continue to improve and stabilize around 15 chords. After this point, adding more chords has little impact on performance.

It is also evident that 3-tokens representation outperforms the 1-token representation in both long short-term networks and gated recurrent units, while in the vanilla recurrent neural networks the performances are swapped. The last plot shows the length distribution of testing sequences, witch aligns closely with the part size distribution shown in Figure 8.3.5.

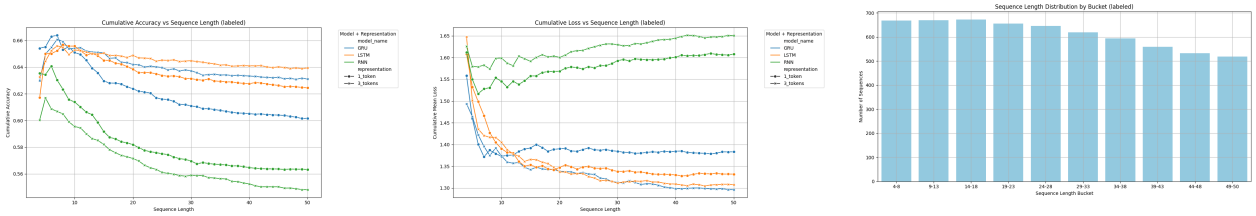
In Figures 9.3.4 and 9.3.5, the 3-tokens representation again outperforms the 1-token one for long short-term memory networks and gated recurrent units, while the opposite trend is observed for the vanilla recurrent networks. The evaluations behind these figures were conducted in the labeled and filtered datasets, respectively. Compared to the segmented dataset, the sample distributions in these evaluations are more balanced. Accuracy appears to decrease as models receive larger chord sequences as input, while on the other hand



Cumulative Accuracy (Segmented) | Cumulative Loss (Segmented) | Sequence Length Distribution (Segmented)

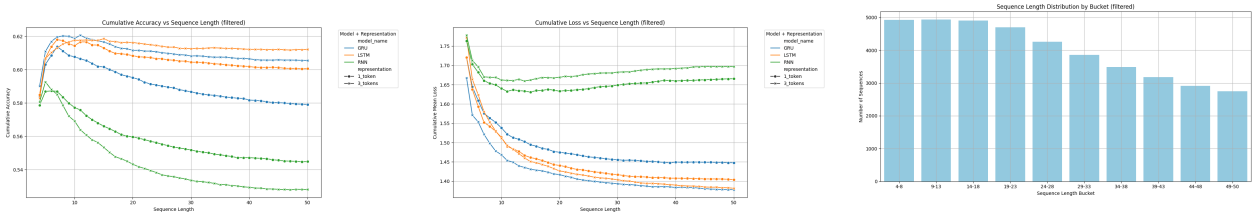
Figure 9.3.3: Cumulative Accuracy, Cumulative Loss, and Sequence Length Distribution for the Segmented dataset.

the loss generally decreases with additional chords, with he recurrent neural networks being once again the primary exception. Overall, the models appear to become more confident but not necessarily more accurate.



Cumulative Accuracy (Labeled) | Cumulative Loss (Labeled) | Sequence Length Distribution (Labeled)

Figure 9.3.4: Cumulative Accuracy, Cumulative Loss, and Sequence Length Distribution for the Labeled dataset.



Cumulative Accuracy (Filtered) | Cumulative Loss (Filtered) | Sequence Length Distribution (Filtered)

Figure 9.3.5: Cumulative Accuracy, Cumulative Loss, and Sequence Length Distribution for the Filtered dataset.

For better understanding on the model performance, we recorded the most common, the most hurtful, and the most impactfull mistakes across all models during evaluation.

Table 9.3 lists the most common prediction mistakes. Many chord pairs appear in both directions (A→B and B→A). The "mean added loss" refers to the average increase in loss when a chord is mispredicted, meaning how smaller would the total loss be if the true chord had been predicted for this instance. The mean added losses range from 1.22 to 1.88.

In Table 9.4 we show the most hurtful mistakes, representing rare errors that produce large losses when they occur. It is interesting to note that these chords are outliers, being mistaken once in the whole evaluation while creating a huge loss impact. The mean added loss ranges from 35.35 to 65.10.

Finally, Table 9.5 presents the most impactful mistakes overall combining frequency and severity. Most of the entries in this top 20 also appear amongst the most common mistakes and their mean added loss ranges

from 1.23 to 1.93.

Table 9.3: Top 20 Most Common Chord Mistakes

True	Pred	Total Count	Mean Added Loss
C	G	13889	1.31
G	D	11526	1.23
G	C	11393	1.29
D	G	11369	1.31
D	A	9320	1.26
Emin	G	9032	1.81
F	C	8495	1.47
Amin	C	8467	1.63
C	D	7513	1.32
F	G	7144	1.46
A	D	7050	1.28
Amin	G	6905	1.88
D	C	6755	1.34
E	A	6729	1.37
C	F	6725	1.28
G	F	6584	1.22
G	A	6559	1.43
C	Amin	6128	1.34
Emin	C	5830	1.80
A	E	5785	1.19

Table 9.4: Top 20 Most Hurtful Chord Mistakes per Instance

True	Pred	Total Count	Mean Added Loss
Fs/C	C7	1	65.10
Eb/A	Fsmin	1	56.38
Adim/C	Gmin	1	49.49
F/B	Dmin	1	48.71
Gs/B	G	1	42.50
Cmin7/A	Eb	1	41.86
Gb11	G	1	41.79
Gsus4/G	Fmaj7	1	40.59
Csmin/D	Csmin	2	39.73
Cminmaj7	Asno3d	1	39.05
Fsus4/E	Fs	1	38.95
Gbminadd13	Dmin	1	38.08
Aminadd13/E	E	1	37.04
Fsmin11/A	A	1	36.99
Bbsus4/G	Eb	1	36.82
Bb/As	Emin	1	36.57
Fmaj9/C	Cmaj7/G	1	36.35
Cdim/Eb	F7	1	36.29
Gsmin13	Ddim7	1	36.25
Emin9/G	Badd9/Ds	1	35.34

Table 9.5: Top 20 Most Impactful Chord Mistakes Overall

True	Pred	Total Count	Mean Added Loss	Total Added Loss
C	G	13889	1.31	17854
Emin	G	9032	1.81	16129
D	G	11369	1.31	14536
G	C	11393	1.29	14409
G	D	11526	1.23	13964
Amin	C	8467	1.63	13744
Amin	G	6905	1.88	13053
F	C	8495	1.47	12179
D	A	9320	1.26	11466
Emin	C	5830	1.80	10193
F	G	7144	1.46	10078
C	D	7513	1.32	10002
Bmin	D	5537	1.80	9982
Emin	D	5171	1.93	9973
D	C	6755	1.34	9157
E	A	6729	1.37	9089
G	A	6559	1.43	8949
A	D	7050	1.28	8929
C	F	6725	1.28	8597
C	Amin	6128	1.34	8258

Chapter 10

Conclusion

10.1 Discussion

This chapter provides interpretations of the results of both the descriptive analyses and the classification and predictive experiments in the context of the research objectives outlined in chapter proposal. This study aimed to analyze the structure of the Chordonomicon dataset and investigate whether interpretable models can capture meaningful patterns in symbolic music compared to more complex machine learning methods and whether lightweight models can learn temporal dependencies while also being aware of part-level structure and having more expressive representation of chord, the decomposed form. By combining large-scale statistical exploration with sequential modeling, the analyses provide insights into both the shared backbone of tonal music and the context-dependent variations that distinguish genres, decades, and song sections. The discussion is therefor split into two complementary parts, an examination of the dataset's structural and harmonic properties, and interpretation of the experimental results, linking observed patterns to model performance.

Descriptive Analysis

The Chordonomicon reveals that popular music is governed by strong harmonic regularities alongside subtle stylistic choices. Songs generally contain specific number of parts, 6-8 in average, with individual sections averaging around 8 chords in length, with local maxima often appearing multiples of four giving information about common chord progression lengths and reflecting conventional metric phrasing. Segment-wise, verses and choruses dominate in both frequency and length showcasing their importance in song structure and flow, while intros, outros and other parts are brief and serve more as transitions rather than containing the essence of the musical piece. Metal songs tend to have longer parts, Rap as well compensating for having less parts on average, highlighting genre-specific compositional choices withing a shared formal grammar. Part length and structure in general are remarkably stable across genres and decades, indicating that the underlying song framework is highly standardized.

Considering harmonic vocabulary, major and minor triads overwhelmingly dominate (>90%), while irregular ones are concentrated in Metal, Rock and Punk. Extensions, expanding the complexity of the chord by using more pitches, showcase preferences in Jazz and Soul to be more expressive, while the primarily stylistic usage of inversions is also focused on them despite being extremely rare. Over time, the dataset shows a gradual simplification of music, all the above characteristics seem to shrink approaching recent years, reflecting shifts in production techniques, aesthetic priorities, and trend pressures.

Focusing on sequential features and dependencies, root motions closely follow the circle of fifths, distances of 5 semitones are more likely to happen, followed by the 2 semitone movements, while transitions of 6 semitones are the most scarce. These observations align with the most common chord progressions from music theory such as I-IV-V-I, which is a sequence of 5-2-(-5) semitone distances. Transition directions are also balanced, showing no preference on upward or downward motions. These patterns are consistent across genres and decades, suggesting tonal stability as a backbone of popular music.

Furthermore, examining longer contextual behaviors, n-gram analyses reveal common progression pattern and that chord transitions vary when between structural sections. By measuring the total variation distance moving from overall chord transitions to ones occurring on part boundaries, for bigrams and trigrams, it is evident that harmonic rules based on the context change across song segments. While it is well known that song sections provide contextual changes in lyrics and melody, our results show that harmonic context itself also changes systematically. Local chord sequences are constrained by part boundaries, reinforcing the idea that part sizes often reflect complete harmonic phrases and highlighting the importance of incorporating structural context in predictive modeling of symbolic chord sequences.

Extending the contextual range to 4-,5- and 6-grams, we employed latent distance models to represent songs as graph through a careful designing process that aims to highlight genre and decade specific patterns through clustering. By embedding n-grams and categories into a shared latent space, the model visualizes the statistical proximity between genres and decades, providing a more interpretable lens on these sequential patterns. This visualization revealed relations between genres and temporarily adjacent decades by positioning them closely, confirming that n-gram statistics can capture some of the temporal relationships. Notably, the majority of the n-grams occupy dense central regions, reflecting the shared harmonic patterns across all music. This latent space approach provides further evidence that harmonic foundations are universal, with stylistic and temporal differentiation emerging subtly in higher-order local patterns.

Experiments

Complementing these observations, the sequential modeling experiments on genre and decade classification and chord prediction exhibit interesting results.

For the classification tasks, lightweight recurrent based and state-space models, as well as latent distance models, were generally unable to achieve strong predictive performance. Models trained on pitch-class representations alone, performed only moderately, while vanilla recurrent neural networks and latent distance-based representations approached near-chance levels. More sophisticated architectures, such as the gated recurrent units, the long short-term memory networks and Mamba, performed better but still struggled, highlighting the lack of strongly discriminative harmonic cues across genres and decades.

Despite the weak classification results, the experiments provide valuable insights. Confusion matrices reveal, from the sequential models inference, reveal that misclassifications often involve stylistically similar categories, consistent with the proximities observed in the latent space of n-grams. This reinforces the notion that a shared harmonic structure extends throughout the music genres and decades, enriching our understanding of music in a harmonic level as a system with stable foundations and subtle variations.

The chord prediction experiments further illuminate the role of harmonic context and part level structure. Across all sequential architectures experiments reveal a clear threshold in the relationship between training data size and model performance. As the dataset used for training grows, accuracy and loss improve steeply until around 200,000 chord progression samples. After that threshold gains plateau, indicating a saturation point where the models have already captured the dominant harmonic dependencies present in the corpus, with extra samples providing marginal refinements rather than new information about tonal structure.

Examining sequence length effects further illuminates how much context these models effectively use. Prediction performance improves as longer input sequences provide richer harmonic indications. This benefit saturates after roughly 15 chords, but without degrading performance significantly. This suggests that the predictive information contained in the immediate harmonic neighborhood is finite and mostly concentrated locally. It is interesting that the 3-tokens representation consistently outperforms the 1-token approach, as focusing on each chord component separately, meaning root, quality and extensions, and inversion, allows the models to internalize relationships between chord parts rather than memorizing each chord symbol. This demonstrates that harmonic understanding benefits from representing chords as compositional objects rather than atomic tokens.

Vanilla recurrent neural networks, however, generally fall behind their gated variants. Their limited capacity to retain and update contextual information over longer sequences prevents them from modeling longer-range and more complex harmonic dependencies. Interestingly, the dynamics of the 3-tokens and 1-token approaches are swapped. This degradation arises because uncertainty is independently generated for each

chord component during training. When the model combines the predicted components to construct the full chord, it also accumulates the uncertainties that accompany each one, leading to increased overall error.

Finally, when models are tested on entire songs without respecting part segmentation overall performance degrades, though longer input sequences appear to offer partial benefits. This improvement is rather deceptive. While accuracy tends to decline slightly, the loss continues to improve, indicating that the models become more confident on their predictions. This occurs because once a part boundary is crossed, the harmonic context effectively resets and the tonal rules guiding the flow of one section no longer apply to the next. As a result, sequences that span multiple parts blend incompatible contexts, leading to systematic mistakes. The observed confidence gain emerges from the model gradually adapting to the new local context after transitioning between parts. This reinforces the dependent relationship between harmonic progressions and song segmentations and highlights the importance of modeling part-level structure when working with symbolic chord sequences. Notably, performance differs across the labeled and filtered datasets, reflecting that songs with additional annotation tend to be more curated.

10.2 Limitations and Future Work

While the analysis and experiments presented in this thesis provide valuable insights into the statistical and sequential structure of symbolic chord progressions, the scope and design of this study are accompanied by several limitations. These limitations exhibit both challenges in modeling tonal music and opportunities for extending the work. Below we present the points that not only identify constraints of the current approach but also suggest directions for future research, building on the insights provided by the exploratory data analysis and the classification and predictive modeling pipelines.

- **Contextual Shifts:** Chord prediction suffers when progressions cross part boundaries, highlighting the influence of contextual shifts on harmonic progressions. Future work could explore a complementary Part Detection task, using model uncertainty to identify segment boundaries. Incorporating part-level predictions into chord modeling could help the model detect when local harmonic context resets, improving performance on both tasks.
- **Dataset imbalance and curation:** Although the Chordonomicon dataset is a rich source of musical information, the balancing procedure needed towards classification tasks discarded many entries, potentially losing on important patters and trends. Future studies could investigate methods leveraging imbalanced datasets, such as weighted loss functions or data augmentation directed towards the categories, to avoid losing harmonic and stylistic information.
- **Limitations of symbolic chord information:** Symbolic chord progressions capture harmonic structure but ignore other musical meaningful dimensions, such as melody, rhythm, tempo, and duration. Integrating multimodal features, including audio representations, or additional annotations, could enhance both classification and prediction performance.
- **Chord Mispredictions:** The models' prediction mistakes fall into two broad types, the rare ones, with insignificant appearances, where the models get confused the most resulting in high loss values and the most frequent, musically plausible confusions (for example Cmaj vs Gmaj or Cmaj vs Amin), that also align with the distributions shown in Figure 8.3.14. Future work could incorporate smoothing or loss adjustment techniques that account for musically close alternatives, reducing the impact of acceptable harmonic substitutions on model training.

Chapter 11

Bibliography

[1] Müller, M. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer Verlag, 2015. ISBN: 978-3-319-21944-8. URL:

[2] Berklee Online. *Circle of Fifths: The Key to Unlocking Harmonic Understanding*. Accessed: 2025-10-23. Berklee College of Music. Feb. 2023. URL:

[3] Elman, J. L. "Finding structure in time". In: *Cognitive science* 14.2 (1990), pp. 179–211.

[4] Bengio, Y., Simard, P., and Frasconi, P. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 5 (Feb. 1994), pp. 157–66. DOI: 10.1109/72.279181.

[5] Hochreiter, S. and Schmidhuber, J. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[6] Can, T., Krishnamurthy, K., and Schwab, D. J. "Gating creates slow modes and controls phase-space complexity in grus and lstms". In: *Mathematical and Scientific Machine Learning*. PMLR. 2020, pp. 476–511.

[7] Hoff, P. D., Raftery, A. E., and Handcock, M. S. "Latent space approaches to social network analysis". In: *Journal of the american Statistical association* 97.460 (2002), pp. 1090–1098.

[8] Handcock, M. S., Raftery, A. E., and Tantrum, J. M. "Model-based clustering for social networks". In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 170.2 (2007), pp. 301–354.

[9] Sarkar, P. and Moore, A. W. "Dynamic social network analysis using latent space models". In: *Acm sigkdd explorations newsletter* 7.2 (2005), pp. 31–40.

[10] Krivitsky, P. N. et al. "Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models". In: *Social networks* 31.3 (2009), pp. 204–213.

[11] Nickel, M. and Kiela, D. "Poincaré embeddings for learning hierarchical representations". In: *Advances in neural information processing systems* 30 (2017).

[12] Harte, C. et al. "Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations." In: *ISMIR*. Vol. 5. 2005, pp. 66–71.

[13] Kantarelis, S. et al. "Functional harmony ontology: Musical harmony analysis with Description Logics". In: *Journal of Web Semantics* 75 (2023), p. 100754.

[14] Kantarelis, S. et al. "CHORDONOMICON: A Dataset of 666,000 Songs and their Chord Progressions". In: *arXiv preprint arXiv:2410.22046* (2024).

[15] Kanat, A. C. *Latent Distance Model (LDM)*. Accessed: 2025-11-02. 2023.

[16] musictheory.net. *Scale Degrees*. Accessed: 2025-10-22. 2025. URL:

[17] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

[18] Bergmann, D. *What is machine learning?* Accessed: 20 October 2025. 2025.

[19] Lee, J. and Lee, J.-S. "Music popularity: Metrics, characteristics, and audio-based prediction". In: *IEEE Transactions on Multimedia* 20.11 (2018), pp. 3173–3182.

[20] Marxer, R. and Purwins, H. "Unsupervised incremental online learning and prediction of musical audio signals". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.5 (2016), pp. 863–874.

[21] China, C. R. *Five Machine Learning Types to Know*. Accessed: 2025-10-20. 2025.

[22] Wikipedia contributors. *Neural network (machine learning) — Wikipedia, The Free Encyclopedia*. [Online; accessed 20-October-2025]. 2025. URL:
[23] Lee, F. *What Is a Neural Network?* Accessed: 2025-10-20. 2025.
[24] Keim, R. *How to Train a Basic Perceptron Neural Network*. Accessed: 2025-10-23. Nov. 2019. URL:
[25] Korzeniowski, F. and Widmer, G. *Feature Learning for Chord Recognition: The Deep Chroma Extractor*. 2016. arXiv: 1612.05065 [cs.SD]. URL:
[26] Han, Y., Kim, J., and Lee, K. "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.1 (Jan. 2017), pp. 208–221. ISSN: 2329-9304. DOI: 10.1109/taslp.2016.2632307. URL:
[27] Wikipedia contributors. *Rectified linear unit — Wikipedia, The Free Encyclopedia*. [Online; accessed 22-October-2025]. 2025. URL:
[28] Cheng, C. "Principal Component Analysis (PCA) Explained Visually with Zero Math". In: *Towards Data Science* (2022). Accessed: 2025-10-25. URL:
[29] Mitchell, T. "What are Embedding Models? An Overview". In: *Couchbase Blog* (2024). Accessed: 2025-10-25. URL:
[30] Ixnay. *Recurrent Neural Network Unfold*. CC BY-SA 4.0, Wikimedia Commons. n.d.
[31] Ixnay. *Long Short-Term Memory*. CC BY-SA 4.0, Wikimedia Commons. 2017.
[32] Arpit, D. et al. "h-detach: Modifying the LSTM gradient towards better optimization". In: *arXiv preprint arXiv:1810.03023* (2018).
[33] Zhao, J. et al. "Do RNN and LSTM have long memory?" In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11365–11375.
[34] Rizakis, M. et al. "Approximate FPGA-based LSTMs under computation time constraints". In: *International Symposium on Applied Reconfigurable Computing*. Springer. 2018, pp. 3–15.
[35] Cho, K. et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).
[36] Ixnay. *Gated Recurrent Unit*. CC BY-SA 4.0, Wikimedia Commons. 2017.
[37] Vaswani, A. et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL]. URL:
[38] Gu, A., Goel, K., and Ré, C. *Efficiently Modeling Long Sequences with Structured State Spaces*. 2022. arXiv: 2111.00396 [cs.LG]. URL:
[39] Gu, A. and Dao, T. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. 2024. arXiv: 2312.00752 [cs.LG]. URL:
[40] Eck, D. and Schmidhuber, J. "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks". In: *Proceedings of the 12th IEEE workshop on neural networks for signal processing*. IEEE. 2002, pp. 747–756.
[41] Garoufis, C., Zlatintsi, A., and Maragos, P. "An LSTM-based dynamic chord progression generation system for interactive music performance". In: *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2020, pp. 4502–4506.
[42] Han, X., Chen, F., and Ban, J. "Music emotion recognition based on a neural network with an inception-gru residual structure". In: *Electronics* 12.4 (2023), p. 978.
[43] Huang, C.-Z. A. et al. "Music transformer". In: *arXiv preprint arXiv:1809.04281* (2018).
[44] Huang, Y.-S. and Yang, Y.-H. "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions". In: *Proceedings of the 28th ACM international conference on multimedia*. 2020, pp. 1180–1188.
[45] Krebs, F., Böck, S., and Widmer, G. "An Efficient State-Space Model for Joint Tempo and Meter Tracking." In: *ISMIR*. 2015, pp. 72–78.
[46] Tang, H. et al. "Spatial-Temporal Graph Mamba for Music-Guided Dance Video Synthesis". In: *IEEE transactions on pattern analysis and machine intelligence* (2025).
[47] Pyrovolakis, K., Tzouveli, P., and Stamou, G. "Multi-modal song mood detection with deep learning". In: *Sensors* 22.3 (2022), p. 1065.