

ACOUSTIC SEGMENT MODELING AND PREFERENCE RANKING FOR MUSIC INFORMATION RETRIEVAL

A Dissertation
Presented to
The Academic Faculty

by

Jeremy T. Reed

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2010

ACOUSTIC SEGMENT MODELING AND PREFERENCE RANKING FOR MUSIC INFORMATION RETRIEVAL

Approved by:

Professor Mark Clements,
Committee Chair
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Chin-Hui Lee, Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor David Anderson
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor William Hunt
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Parag Chordia
School of Music
Georgia Institute of Technology

Date Approved: 15 October 2010

To Jenny,

for maintaining your sanity, despite my best efforts.

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Professor Chin-Hui Lee, for allowing me to explore an area that was novel to us both. This dissertation would not have been possible without his excitement, expertise, and guidance. The lessons learned over the past few years will carry beyond just my career. I would also like to thank the members of my committee for their insight and support: Dr. David Anderson, Dr. Parag Chordia, Dr. Mark Clements, and Dr. William Hunt. In addition, I owe gratitude to Dr. Jim McClellan, Dr. Russ Mersereau, and Dr. Aaron Lanterman for appointing me as the head teaching assistant for ECE 2025: Introduction to Signal Processing, which is where I learned the importance of teaching skills and communication.

I would also like to thank Dr. Shigeki Sagayama for affording me the opportunity to work with his group at The University of Tokyo. I owe respect and thanks to his students and staff for making me feel at home in their lab and a full member of their group during my stay. I would like to especially thank Yushi Ueda and Uchiyama Yuki for their contributions on our project.

I am particularly grateful for colleagues that I have collaborated with on various projects, assignments, and studies: Byungki Byun, Ilseo Kim, Jinyu Li, Chengyan Ma, Brett Matthews, Antonio Moreno, Sabato Marco Siniscalchi, Yu Tsao, and Sibel Yaman. In addition, I would like to thank my friends for their support over these last few years: Brandon Beacher, Jim and Kelly Garrison, Jason McInnis, Aaron Nowak, Chris and Kim Swenson, and Joey and Sara Wallace.

I am indebted to my parents, Larry and Geri, for their love, support, and nurturing my inquisitive mind. To my brothers and sisters, I am thankful for your support and

encouragement. I would be speciest if I did not thank Marley, my dog, for his comfort and playful exuberance. Finally, I would like to thank my fiancée, Jenny Matthews, whose love and encouragement has brightened the darkest days. I am grateful to have found you because, as I am sure will usually be the case, I was wrong and you were right - there is one person who is our match. I cannot wait to start our lives together.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
SUMMARY	xii
I INTRODUCTION	1
II BACKGROUND	5
2.1 Recommendation Technologies for Music Information Retrieval	5
2.1.1 Collaborative-Filtering Systems	6
2.1.2 Content-Based Systems	7
2.1.3 Hybrid and Fuzzy Approaches	14
2.2 Automatic Speech Recognition	15
2.2.1 Feature Extraction	15
2.2.2 Acoustic Modeling	16
2.2.3 Language Modeling	18
2.2.4 Recognition	18
2.3 Defining Musical Similarity	19
2.4 Ordinal Regression	23
III ACOUSTIC SEGMENT MODELING FOR MUSIC INFORMATION RE- TRIEVAL	26
3.1 Feature Representation	29
3.2 Initial Segmentation and Transcription	29
3.3 ASM/HMM Training	32
3.4 Experiments	32
3.4.1 Front-end Specifications	33
3.4.2 Back-end Classifier	36

3.4.3	Database and Evaluation	40
3.4.4	Results	41
3.5	Summary	43
IV	MUSIC STRUCTURE DETECTION	46
4.1	Temporal Tag Identification	46
4.1.1	Semantic Multimedia Tags	46
4.1.2	Acoustic Segment Modeling for Tag Identification	48
4.1.3	Baseline Tag Identification Algorithm: Mixture-of-Hierarchies	49
4.1.4	Results	51
4.2	Musical Chord Recognition	55
4.2.1	Chord Recognition Features	56
4.2.2	ASMs to Chords	58
4.2.3	Baseline Chord Recognition System	60
4.2.4	Data and Evaluation	60
4.2.5	Results	61
4.3	Summary	62
V	MUSIC SIMILARITY FOR CONTENT-BASED ALGORITHMS	65
5.1	Song Description Vector	66
5.2	Attribute-based Taxonomy	68
5.3	Discriminative-Training Song Classification	69
5.4	Experimental Results	70
5.5	Summary	72
VI	CONTENT-BASED PREFERENCE RANKING	75
6.1	Discriminative-Training Ordinal Regression	76
6.2	Baseline Ranking Algorithm: PRank [29]	78
6.3	Data Modeling	79
6.3.1	Acoustic Segment Modeling Specifications	79
6.3.2	Spectral-based approach 1: Single Gaussian	80

6.3.3	Spectral-based approach 2: Bag-of-Timbres	80
6.3.4	Spectral-based approach 3: GMM Supervectors	81
6.4	Evaluation Metrics	84
6.5	Experiment 1: Explicit Artist Prediction	85
6.6	Experiment 2: Implicit Song Rankings	90
6.7	Summary	92
VII	IMPROVEMENTS AND EXTENSIONS	94
7.1	Incorporating Multiple Acoustic Features	95
7.1.1	Extending ASM to Harmonic Information	95
7.1.2	Database	96
7.1.3	Results	97
7.2	Hybrid Approach	98
7.2.1	Collaborative-Filtering Baseline	99
7.2.2	Content-Boosted Collaborative Filtering	100
7.2.3	Database and Evaluation Metrics	101
7.2.4	Results	102
7.3	Summary	102
VIII	CONCLUSION	104
	REFERENCES	108

LIST OF TABLES

3.1	Convergence of ASM procedure.	34
3.2	Accuracy versus iteration number.	42
3.3	Confusion matrix for the Magnatunes dataset, given as counts. The last row and column are the precision and recall percentages, respectively.	42
3.4	Comparison of genre classification accuracy on the GTZAN dataset between the ASM procedure and selected published results.	43
3.5	Genre classification accuracy when using an ASM vocabulary of 64 and 128.	43
4.1	Results for each tag in terms of EER for the proposed (Prop) and baseline (Base) approaches. Tags are labeled as either temporally-based, globally-based, or both, as indicated in the parenthesis. Bold face indicates McNemar statistical significance.	53
4.2	Retrieval mean average precision for the acoustic segment modeling and baseline approaches.	54
4.3	Example ASM sequences for chords.	58
4.4	Isolated chord accuracies for the first training and testing set.	62
4.5	FER for baseline and ASM approach.	63
4.6	HTK results for the chord detection task.	63
5.1	Confusion matrix for classifying genres by Pandora attributes.	72
6.1	Comparison of ratings distribution for the Amazon dataset in [140] and the Yahoo! dataset.	84
6.2	Results for different system configurations with 250 training vectors. See text for details.	86
6.3	Results the ASM/DTRANK system using different training sizes.	87
6.4	Results for ratings prediction on implicit ratings	92
7.1	Results using MFCCs only, PCPs only, and MFCCs and PCPs.	97
7.2	Tag annotation EER for the PCP-based and MFCC-based ASM systems.	98
7.3	Results of three hybrid systems compared to a purely collaborative-filtering (CF) approach	102

LIST OF FIGURES

2.1	Content-based algorithm.	8
2.2	Representations of song similarity. Each rectangle represents a single song, and arrows are drawn between the closest songs given the representation. In (a), songs have a static texture (uniform color). In (b), temporal features are extracted, but songs are still represented as a single entity. In (c), songs are tokenized into segments so that semantic information may be considered.	9
2.3	A hidden Markov model for the phoneme /aa/.	17
2.4	Eric Clapton spans multiple styles and genres.	19
2.5	Tags for <i>Soul Meets Body</i> by Death Cab for Cutie from social networking site last.fm.	22
2.6	Tags for <i>Soul Meets Body</i> by Death Cab for Cutie from Pandora.	22
2.7	Differences between multi-class classification, regression, and ordinal regression.	25
3.1	Modeling approach for automatic speech recognition.	27
3.2	Modeling approach for music information retrieval.	28
3.3	Acoustic segment modeling block diagram.	28
3.4	Initial segmentation of a song segment in the GTZAN dataset.	31
3.5	Example ASM transcript.	31
3.6	Block diagram for ASM genre classification.	33
3.7	Bigram frequency versus ranking applied to the MIREX dataset.	35
3.8	Support vector machine for a 2-class problem. See text for details.	39
4.1	Diagram of mixture-of-hierarchies algorithm used in [125]. Weights between a given song and a given tag represent the salience of the tag in the song.	50
4.2	Example of ASM tokenization for two similar melodies as a solo (lower waveform) and with polyphonic ornamentation (upper waveform).	55
5.1	Sigmoid-based loss function.	70
5.2	Attribute weights before and after the discriminative-training procedure for a chosen cluster.	71
5.3	Empirical loss for the discriminative-training song description clusters.	71

6.1	Illustration on the lack of calibration when using GMMs to model two objects individually. Blue, solid circles are mixtures for song A, and yellow, textured circles are mixtures for song B.	82
6.2	Rationale of the supervector approach. Each mixture of an artist or song GMM is an adapted mixture from a universal model.	82
6.3	GMM-based supervector approach.	83
6.4	Estimated empirical ranking loss in (6.1) and true ranking loss in (2.6) averaged over the first ten users.	88
6.5	Values of the 0-1 loss approximation in (6.4) at the boundary between $r = 3$ and $r = 4$ for each training sample of a typical user at the first and last iteration.	89
6.6	Misclassification measure given in (6.5) for iterations 10 and 2000 for a random user.	90
7.1	Front-end for combined MFCC/PCP approach.	96

SUMMARY

This dissertation focuses on improving content-based recommendation systems for music. Specifically, progress in the development in music content-based recommendation systems has stalled in recent years due to some faulty assumptions:

1. most acoustic content-based systems for music information retrieval (MIR) assume a bag-of-frames model, where it is assumed that a song contains a simplistic, global audio texture
2. genre, style, mood, and authors are appropriate categories for machine-oriented recommendation
3. similarity is a universal construct and does not vary among different users

The main contribution of this dissertation is to address these faulty assumptions by describing a novel approach in MIR that provides user-centric, content-based recommendations based on statistics of acoustic sound elements. First, this dissertation presents the acoustic segment modeling framework that describes a piece of music as a temporal sequence of acoustic segment models (ASMs), which represent individual polyphonic sound elements. A dictionary of ASMs generated in an unsupervised process defines a vocabulary of acoustic tokens that are able transcribe new musical pieces. Next, standard text-based information retrieval algorithms use statistics of ASM counts to perform various retrieval tasks. Despite a simple feature set compared to other content-based genre recommendation algorithms, the acoustic segment modeling approach is highly competitive on standard genre classification databases.

Fundamental to the success of the acoustic segment modeling approach is the ability to model acoustical semantics in a musical piece, which is demonstrated by the detection of musical attributes on temporal characteristics. Further, it is shown that the acoustic segment modeling procedure is able to capture the inherent structure of melody by providing near state-of-the-art performance on an automatic chord recognition task.

This dissertation demonstrates that some classification tasks, such as genre, possess information that is not contained in the acoustic signal; therefore, attempts at modeling these categories using only the acoustic content is ill-fated. Further, notions of music similarity are personal in nature and are not derived from a universal ontology. Therefore, this dissertation addresses the second and third limitation of previous content-based retrieval approaches by presenting a user-centric preference rating algorithm. Individual users possess their own cognitive construct of similarity; therefore, retrieval algorithms must demonstrate this flexibility. The proposed rating algorithm is based on the principle of minimum classification error (MCE) training, which has been demonstrated to be robust against outliers and also minimizes the Parzen estimate of the theoretical classification risk. The outlier immunity property limits the effect of labels that arise from non-content-based sources. The MCE-based algorithm performs better than a similar ratings prediction algorithm. Further, this dissertation discusses extensions and future work.

CHAPTER I

INTRODUCTION

Since the late 1990s, music has become increasingly cheaper to create, distribute, and market. As a result, music production has skyrocketed, with the number of albums produced between 2002 and 2007 more than doubling. Despite this tremendous growth in production, total revenue has decreased even though digital downloads have increased dramatically [117].

The lack of revenue growth, in part, can be attributed to a decreased need on the part of consumers to possess music. Not only has it become easier to obtain free music from both legal and illegal sources [106], but the proliferation of wireless multimedia players and online radio stations has more consumers streaming music from the cloud. Therefore, there is no longer a need to possess music as either physical media or digitally encoded bits. As such, the music industry is shifting (regretfully) from a physical production industry to a subscriber industry. New online radio stations, like Pandora¹ and last.fm², fit this mold and can find parallels in other industries, e.g., Netflix³ in the movie industry. However, despite their similarities, these and other online distribution channels have been shown in [25] to differ from their movie industry cousin by one key factor: the inability to tap *The Long Tail* [1].

Described by Chris Anderson [1], The Long Tail theorizes that as distribution channels become less restricted, less popular items will contribute more toward revenue under limited distribution channels. Unlike brick-and-mortar stores, Internet-based stores and distributors such as Netflix are not limited by shelf space. Therefore,

¹<http://www.pandora.com>

²<http://www.last.fm>

³<http://www.netflix.com>

people are able to find their individual niche market whereas traditional stores must cater to the masses to maximize their revenue. To tap into The Long Tail, Netflix and similar websites rely on recommendation technology to inform users about possible titles they might not know or remember. In fact, as much as 60% of rentals on Netflix are due to recommendations [25]. This fact even spurred Netflix to award a million dollar prize if one could beat Netflix's recommendation algorithm by 10% [13].

One key component of identifying the proper use of recommendation technologies is market penetration into The Long Tail; however, only 1% of music titles account for the vast majority of sales [117]. Therefore, it is likely that music recommendation algorithms are currently unable to penetrate niche markets. This can also be seen by the fact that as much as 60% of the existing music on iPods has never been played [25]. In effect, people are currently buying the wrong music for their personal tastes. The winning technologies for the music industry will not only need the correct price structure, but also possess the ability to give highly personalized recommendations.

Recommendation algorithms are grouped into two types: collaborative-filtering and content-based. Collaborative-filtering algorithms rely on identifying groups or items with similar behaviors, e.g., "People who have listened to *A* also listen to *B*." However, collaborative-filtering algorithms have difficulty recommending new or rare content and often reward popular items to create a popularity gap. Content-based algorithms extract information from the audio signal directly and can generate recommendations for any piece of music for which audio is available. However, since most music information retrieval (MIR) approaches are imperfect, such as genre classification, these recommendations are often noisy.

This dissertation focuses on improving content-based systems for music recommendation. Specifically, progress in the development of content-based music recommendation algorithms has stalled in recent years due to three faulty assumptions:

1. most acoustic content-based recommendation algorithms utilize a bag-of-frames model, where it is assumed that a song contains a simplistic, global audio texture.
2. genre, style, mood, and authors are appropriate categories for machine-oriented recommendation.
3. similarity is a universal construct and does not vary among different users.

This dissertation describes a novel approach in MIR that develops user-centric, content-based recommendations. Specifically, the three faulty assumptions in current content-based music recommendation algorithms are examined and solutions are proposed. First, a musical piece is described as a temporal sequence of acoustic segment models (ASMs), where each ASM describes the short-time temporal structure of an acoustic element or sound. A dictionary of ASMs generated in an unsupervised process defines a vocabulary of sounds to transcribe new pieces of music. Standard text-based information-retrieval algorithms can then provide recommendations. Using the same feature set, it is shown that the acoustic segment modeling approach is superior than a state-of-the-art spectral-based approach, i.e., bag-of-frames. Fundamental to the success of the ASM approach is the ability to model acoustical semantics in a piece of music, which is demonstrated by the detection of musical attributes that contain temporal characteristics. Second, this dissertation demonstrates that some classification tasks, such as genre, possess information that is not contained in the acoustic signal; therefore, attempts at modeling these categories using the acoustic content is ill-fated. Therefore, a user-centric ratings prediction algorithm is proposed to allow for individual users to possess their own cognitive construct of similarity.

This dissertation is organized as follows. Chapter 2 details the necessary background for this dissertation by giving an overview of recommendation technologies and defining music similarity. In addition, a basic presentation for automatic speech

recognition (ASR) is presented. The success of ASR technologies is the motivation behind the acoustic segment modeling approach, presented in Chapter 3. While the acoustic segment modeling approach is found to be competitive with existing genre recognition systems, Chapter 4 investigates the temporal modeling advantages obtained with the ASM approach over existing MIR systems. In particular, this is demonstrated on two tasks: semantic tag identification and chord recognition. Further investigation into the genre recognition problem is presented in Chapter 5 and finds that genre is an ill-defined problem for content-based analysis because genre definitions are derived, in part, from factors exterior to the acoustic signal. These results motivate the user-oriented ratings prediction algorithm presented in Chapter 6, which predicts how many “stars” a user will give a song based on the acoustic content. Finally, Chapter 7 presents two extensions for the content-based system presented in this dissertation. The first incorporates multiple sources of acoustic information (i.e., feature types), and the second incorporates collaborative information in a hybrid approach.

CHAPTER II

BACKGROUND

This chapter describes the necessary background for this dissertation. First, an overview of recommendation technologies for music is given. Next, a brief overview of ASR, which serves as the motivation for acoustic segment modeling, is presented. Issues in defining similarity for music are then highlighted. Finally, this chapter concludes with an overview of ordinal regression approaches.

2.1 Recommendation Technologies for Music Information Retrieval

Originally, music recommendation and categorization was performed by a group of experts, such as radio disc jockeys, music-review writers, or a musically knowledgeable friend. This paradigm flourished because specific distribution channels were limited to the physical dimensions of shelf space, number of radio stations, etc. However, as the Internet has grown and created infinite shelf space [1], expert-based recommendations are unable to cope with the diversity and quantity of music. As such, the need for automatic music recommendation systems has increased. Generally, there are two types of automatic recommendation systems: collaborative-filtering and content-based. It should be noted that the boundary between these two classes of recommendation systems is often blurred. In addition, some researchers have proposed hybrid approaches in an attempt to remove the weaknesses of collaborative-filtering systems and content-based systems.

2.1.1 Collaborative-Filtering Systems

When organized by the entity from which correlations are measured, collaborative-filtering algorithms are grouped into two sub-types. User-based algorithms assume users with similar patterns of behavior will remain similar in the future. As an example applied to document retrieval, a user-based system might identify a “neighborhood” for a particular user called the “active user” by finding other users who have viewed the same documents as the active user (called “neighbors”). Documents that have not been viewed by the active user, but have been viewed by many of the neighbors would be returned to the active user in a ranked list. One of the first collaborative-filtering systems applied to the music domain was Ringo, developed by Shardanand and Maes [112]. Users subscribed to Ringo by e-mail and received a list of artists to rate. For artists novel to the active user, predictions were made by a weighted average of other users’ ratings, where the weights were determined by the similarity between the active user and the other users on music they both rated. A ranked list was then returned to the active user.

Conversely, an item-based collaborative-filtering algorithm identifies items that have a high probability of co-occurring. For example, a user who buys a book about the life of Bob Dylan may be recommended a book about Simon and Garfunkel, since a high number of people are interested in books about 1960s folk musicians. A successful example of an item-based collaborative-filtering algorithm is Amazon [69], which has developed a real-time implementation by noting that item-to-item similarity can be computed offline.

An important aspect of collaborative-filtering algorithms is that no information is extracted from the given item explicitly; however this is the cause for a few weaknesses of the collaborative-filtering approach. First, collaborative-filtering approaches suffer from the cold-start problem [110], where new or rarely used content cannot be

recommended because it contains few, if any, ratings. This creates a feedback cycle where only the popular items are recommended, even if the rare content may be a better fit for the active user. This phenomenon is further aggravated by the fact that collaborative-filtering algorithms average results across many users, which is also called “the wisdom of the crowds” [120]. However, this tends to emphasize items that are already popular and leads to less interesting recommendations. Finally, users may not take the time to rate items explicitly, which leads to data sparsity.

2.1.2 Content-Based Systems

Content-based algorithms obtain information strictly from the item and generate recommendations by identifying items with similar content. For example, a speech recognition algorithm may transcribe the spoken documents in a user’s collection, and a keyword spotting algorithm would identify important terms. Finally, new spoken documents that also contain the same keywords would be recommended to the user. While content-based algorithms can develop a score for every item, they are limited by two assumptions. First, one must be able to model the content correctly. For example, it is important that the automatic speech recognition technology be robust enough to handle the different recording conditions. Second, content-based algorithms rely on the assumption that all useful information for retrieval purposes is contained in the item. For example, a student using a spoken document retrieval database would find it difficult to obtain a list of audio books that are commonly used for summer reading if only keywords in the oral text are considered. This is because the search is about how the information is used and not based on the information contained in the objects.

Regardless of the specific task, content-based recommendation and classification is generally carried out in a two-step process, as shown in Figure 2.1. First, low-level acoustic features are extracted from the entire audio signal or from short, overlapping

frames. Next, a classifier is trained to make a binary or multiple-class decision based on the statistics of the features. For example, in a multiple-class, single-label problem (e.g., genre recognition, artist identification, etc.), the goal is to find

$$\hat{\mathcal{C}} = \arg \max_{\mathcal{C} \in \mathcal{C}} P(\mathcal{C} | \mathcal{X}), \quad (2.1)$$

where $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{|\mathcal{C}|})$ is the set of class labels under consideration, $|\mathcal{C}|$ is the number of distinct classes considered, and \mathcal{X} is the representation of the acoustic signal. Using Bayes’s rule, (2.1) becomes

$$\hat{\mathcal{C}} = \arg \max_{\mathcal{C} \in \mathcal{C}} \frac{P(\mathcal{X}|\mathcal{C}) P(\mathcal{C})}{P(\mathcal{X})} = \arg \max_{\mathcal{C} \in \mathcal{C}} P(\mathcal{X}|\mathcal{C}) P(\mathcal{C}), \quad (2.2)$$

where the denominator, $P(\mathcal{X})$, is dropped because it does not affect the decision. In most applications and for the rest of this dissertation, the prior for the label, $P(\mathcal{C})$, is distributed uniformly and dropped from consideration, unless otherwise stated.

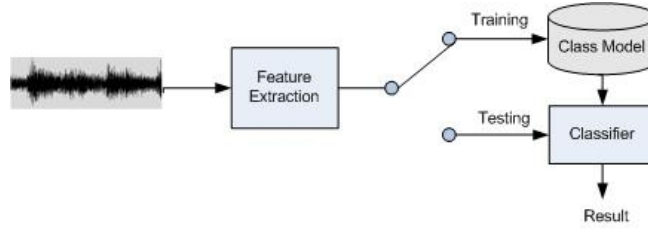
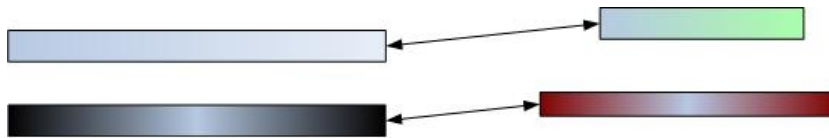


Figure 2.1: Content-based algorithm.

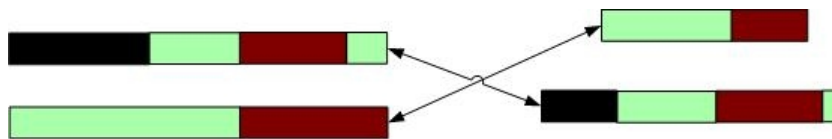
From (2.2), it can be seen that the representation of the likelihood of the data given the class label, $P(\mathcal{X}|\mathcal{C})$, is the most common variant in current MIR approaches. Specifically, approaches to content-based music classification can be categorized by their representation of the data. The first approach assumes a given piece of music contains a static, global texture and is called the “bag-of-frames” approach [3]. The second improves on the bag-of-frames approach by modeling the dynamics within a given song. The final approach treats a song as a concatenation of a shared set of tokens in a vocabulary and is the framework investigated in this dissertation.



(a) Static, global



(b) Dynamic, global



(c) Tokenization

Figure 2.2: Representations of song similarity. Each rectangle represents a single song, and arrows are drawn between the closest songs given the representation. In (a), songs have a static texture (uniform color). In (b), temporal features are extracted, but songs are still represented as a single entity. In (c), songs are tokenized into segments so that semantic information may be considered.

2.1.2.1 *Static, Global Texture*

Assuming a static, global texture assumes that a piece of music is a uniform entity and isolated from other musical works, as shown in Figure 2.2(a). Features are extracted across the whole song or in short, overlapping frames that are treated as identically and independently distributed examples of an underlying probability distribution. A system with this behavior was termed a “bag-of-frames” approach by Aucouturier and Pachet [4] because it is similar to the “bag-of-words” model seen in text retrieval [74]. In [4], the distribution of Mel-frequency cepstral coefficients (MFCCs) is modeled with a Gaussian mixture model (GMM) and songs are compared by using a simulated Kullback-Leibler distance. A related approach can be found in [71], where MFCCs from 25 ms frames are clustered into bins using the k -means algorithm. Distances between songs are then compared by the Earth Mover’s Distance [108]. In [127], rhythmic and pitch features were combined with timbre features to differentiate musical genres. It was found that while timbre features in isolation performed better than rhythmic or pitch features in isolation, the combination improved classification overall. However, none of these features model the dynamic nature of music.

2.1.2.2 *Dynamic, Global Texture*

Attempts at modeling dynamic information generally rely on the use of texture windows [127], features designed to model dynamic information, or the use of hidden Markov models (HMMs) [99]. Such a scenario is shown in Figure 2.2(b). Texture windows [127] summarize a contiguous group of shorter analysis windows (each on the order of 25 ms) by their means and variances over a longer context (on the order of one second). Generally, increasing the duration of texture windows improves performance until a duration of approximately one second, at which point, no noticeable improvement is seen [127]. It should be noted that this approach uses the features

extracted from a texture window in place of features extracted from frames; i.e., decisions are not made on each texture window individually. Therefore, these approaches only describe temporal variations in terms of spread observed over time or frequency of change and do not specifically describe the evolution of features over time.

The most common dynamic features are derivatives of static features that are appended to a vector containing the static features. However, it was shown in [5] that simply appending derivative information did not yield significant improvement, regardless of whether a song was modeled using a GMM or an HMM. More advanced features have been derived in an attempt to model the dynamic nature of music. In [88], distances in fluctuating patterns, which describe the amplitude modulation in different frequency bands, are combined with the distances using the approach in [4] to improve performance. Meng [78] presents two approaches that model MFCCs with an autoregressive process: one assuming feature independence and one that assumes correlations exist over time between MFCC features. The autoregressive coefficients serve as features for the final classifier, which is either a GMM or a generalized linear model. It is demonstrated that significant improvement over texture windows is achieved by assuming features are both correlated in time and across dimension. However, the modeling paradigm is the same as in the approaches mentioned in Section 2.1.2.1; i.e., $P(\mathcal{X}|\mathbb{C})$ uses a static classifier, i.e., a classifier that does not have an explicit temporal structure.

Some approaches use classifiers that are specifically designed for temporal modeling, such as HMMs. The use of HMMs is based on the success in ASR [99], where individual phonemes or words are modeled with an HMM to build a shared vocabulary. However, approaches in MIR have not utilized the full power of HMM modeling. In [5], songs were modeled with a single HMM and it was shown to perform no better than using a GMM with the same number of parameters. Scaringella and Zoia model an entire genre with a single four-state HMM [109]. It should be noted that

both approaches model an entire song or genre with a single HMM, which is different from the application of HMMs in ASR, where a given utterance is decoded using several HMMs. Further, in ASR, each ASM represents a token in a shared vocabulary set. Meanwhile, HMM-based approaches to MIR have largely ignored contextual or semantic information.

2.1.2.3 Tokenization Approaches

The previous approaches model a piece of music as an individual entity, rather than a sequence of shared sounds. In this light, the previous approaches have a similar motivation as the spectral-based approaches to spoken language identification. However, it has been demonstrated that in the context of spoken language identification, approaches based on phone modeling improve performance over spectral GMM-based approaches [145]. Similar concepts may apply in the music domain, but research is limited. One exception is the work of West and Cox [134][135], which builds a model of the acoustic space with a decision tree classifier. Songs are classified according to number of frames assigned to each leaf. This is an example of tokenization [145] because a piece of music is broken into contiguous sections and the entire acoustic space is modeled as discrete units. Further, it is the combination of the discrete units that yields the given realization of the music signal.

Approaches using tokenization, such as the acoustic segment modeling procedure presented in Chapter 3, split the likelihood term, $P(\mathcal{X}|\mathbb{C})$, as

$$P(\mathcal{X}|\mathbb{C}) = P(\mathcal{X}|T) P(T|\mathbb{C}), \quad (2.3)$$

where $T = (t_1 t_2 \dots t_{|T|})$ is a given token stream of length $|T|$. The terms $P(\mathcal{X}|T)$ and $P(T|\mathbb{C})$ are termed the *acoustic model* and the *language model* in the speech recognition community¹, respectively. In this dissertation, the language model is

¹Generally, the ASR community uses W instead of T to represent the token stream since tokens are either words or the building blocks of words. Here T is used to emphasize that tokens may be any temporal segment.

generalized to be a *token set model* since it is not applied to speech. As mentioned in Section 2.1.2.2, the standard paradigm in ASR (i.e., HMMs) is not as prevalent in the MIR community. However, some notable exceptions have been implemented, but in the context of music transcription. Raphael [100] uses HMMs to perform automatic segmentation of monophonic recordings for use in score-following. This was extended in [101] to perform transcription for monophonic piano music. In [37], a monophonic melody spotter is implemented and is based on the task of keyword spotting. In the related field of automatic chord transcription, HMMs have been used extensively [91][114].

Despite these uses, HMMs have not been used in many classification and retrieval algorithms. One reason is that the error rate in transcription tasks remains high, which causes errors to propagate into later stages of the system. For example, the most successful chord recognition systems still perform at just under an 80% frame accuracy rate when only the 12 major and minor chords are considered². A further source of difficulty is defining the modeling detail. In ASR, HMMs generally model at the phoneme or word level [99]; however, it is unclear as to the detail needed to effectively model concepts such as musical similarity or genre. This is further complicated by the fact that music is polyphonic and the relations of co-occurring notes are as important as their temporal ordering [59]. Since source separation is an unsolved problem, many approaches tend to use only a subset of possible chords; e.g., the MIREX competition uses only 12 major and minor chords. However, it is unlikely that such a coarse representation would be sufficient for characterizing genre and similarity across all listeners. Further, extending these models is not computationally feasible. For example, using the labels provided by Harte [45], there are over 400 different chord types in just The Beatles catalogue of studio albums when all chord inversions and extensions are considered. Further, most of these chord types appear

²http://www.music-ir.org/mirex/2009/index.php/Audio_Chord_Detection_Results

very infrequently. Finally, it is not clear which qualities in music are important for similarity judgements.

2.1.3 Hybrid and Fuzzy Approaches

Recently, authors have developed systems that combine collaborative-filtering and content-based analysis. A common approach is to use the predictive strength of collaborative-filtering algorithms to guide the training of content-based algorithms. For example, Stenzel and Kamps [118] use playlists to group songs into clusters. Next, acoustic data is used to train a set of binary classifiers, where each classifier gives a decision as to whether a song belongs to the hypothesized cluster. In [141], a three-way aspect model links a user profile to music content in a Bayesian network. Since collaborative-filtering algorithms generate sparse user-item matrices, [77] uses content-based algorithms to populate missing values.

In addition, fuzzy approaches exist, where it is not clear whether the algorithm is a true collaborative-filtering approach or content-based approach. Generally, such algorithms have users generate “semantic tags” that describe the given multimedia. For example, a user may “tag” a piece by Guns ’N Roses with *rock*, *80s*, *awesome guitar riff*, and *driving*. Tags such as *80s* and *driving* are similar to collaborative-filtering approaches in that they describe *when* and *how* the item is used. Meanwhile, *rock* and *awesome guitar riff* contain information having to do with the content. In essence, these approaches are using humans as high-level feature extractors. Hence, many algorithms that use tags are neither truly a collaborative-filtering or content-based approach. In Chapter 7, the content-based approach presented in this dissertation is extended to a hybrid approach.

2.2 Automatic Speech Recognition

The ASR problem, which is the motivation for the acoustic segment modeling procedure described in Chapter 3, is generally viewed as a *maximum a posteriori* (MAP) problem. The goal is to determine the most likely word sequence, \hat{T} , given the acoustic data, O :

$$\hat{T} = \arg \max_T P_{AM}(O|T) P_{LM}(T), \quad (2.4)$$

where $P_{AM}(O|T)$ is the probability of observing the acoustic data, O , given the word sequence, T , and is known as the acoustic model. The language model, $P_{LM}(T)$, represents the prior probability of observing T . There are generally four steps to ASR: feature extraction, acoustic modeling, language modeling, and recognition.

2.2.1 Feature Extraction

The first important step in ASR is to choose a representation for the acoustic data by extracting a set of features over the course of the audio signal. Features are chosen to reduce the dimensionality of the data, which helps in storage and in classification by reducing the “curse of dimensionality” [17]. Most features used in ASR rely on the discrete Fourier transform (DFT) [98], which assumes stationarity. Since speech is a highly non-stationary signal, the signal is divided into short, overlapping frames prior to feature extraction. It should be noted that smaller analysis windows decrease frequency resolution; therefore, a trade-off exists between temporal resolution and frequency resolution [97]. In addition, speech frames are generally weighted with a window (e.g., Hamming) to reduce the effect of the sharp transitions at the boundaries of the speech frames [98].

Many features have been used to describe speech and music. For this dissertation, MFCCs are used, unless otherwise stated. Previous research has demonstrated that MFCCs are superior to many acoustic features for automatic speech recognition [31] and music retrieval [70]. To extract a set of MFCCs from a given speech frame, the

amplitude spectrum is first obtained by using a DFT. Next, a non-linear filterbank approximates the behavior of the auditory system. Generally, the filters are chosen to have a triangular structure in the frequency domain and to have successive filters overlap by half the Rayleigh resolution [18]. Next, a vector of filterbank energies is created and a discrete cosine transform is applied to reduce the dimension and remove correlation. Obtaining the MFCCs over the entire acoustic signal generates the acoustic representation $O = (\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_{|O|})$, where \mathbf{o}_i is the vector of MFCCs for frame i and $|O|$ is the number of frames extracted in O .

2.2.2 Acoustic Modeling

As shown in Figure 2.3, most state-of-the-art ASR technologies use a left-to-right HMM [99] to model a particular word, syllable, phoneme, etc. to capture the temporal nature of speech. Generally, the parameters of the acoustic models are estimated by one of two supervised approaches [142], depending on whether timing information is given in the provided transcriptions. If timing information is provided, each model can be learned separately by isolating the segments of speech corresponding to the particular phoneme or word. This process is known as isolated training. If no timing information is provided, embedded training is utilized, which concatenates an HMM sequence according to the transcription and then iterates between finding the most likely segmentation and model parameters. Both embedded and isolated training estimate the model parameters using maximum likelihood estimation (MLE) [99].

Another important concept in acoustic modeling is adaptation. While, speaker independent data is cheaper to obtain than speaker dependent data, speaker dependent systems perform better when the same amount of training data is used. Therefore, many systems start with a speaker independent system and adapt model parameters using a small amount of speaker dependent data. The two most well-known adaptation techniques in ASR are maximum likelihood linear regression (MLLR) [65]

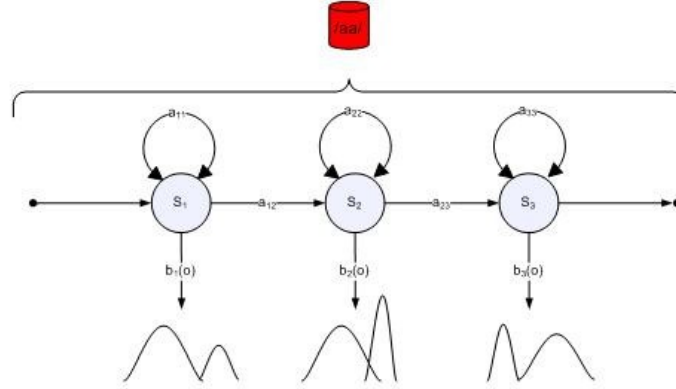


Figure 2.3: A hidden Markov model for the phoneme /aa/.

and MAP [41] adaptation. MLLR estimates a linear transformation matrix from the speaker-dependent adaptation data in order to update the model parameters from a speaker-independent model [65]. MAP adaptation uses a Bayesian approach by estimating a speaker-independent prior and then adapts the parameters with the speaker-dependent data by assuming a conjugate prior [41].

The previously mentioned algorithms estimate a density and are known as generative approaches. By contrast, discriminative-training algorithms attempt to minimize recognition error directly. Maximum mutual information estimation [8] attempts to separate given classes (e.g., phones) by maximizing the posterior probability of the correct class. Minimum classification error (MCE) [54] models the string recognition problem as a problem of minimizing classification risk. Minimum word error and minimum phone error [96] attempt to optimize an approximation for word and phone error rates, respectively. The recently proposed soft margin estimation [67] adapts the concept of soft-margin support vector machines [28] to estimate the HMM parameters.

2.2.3 Language Modeling

The language model dictates the allowable word sequences. For small vocabulary systems, a task grammar that is very restrictive in the types of sequences is easy to implement. For large vocabulary systems, it is difficult to characterize the allowable word sequences that may be encountered; therefore, a probabilistic approach is taken. The state-of-the-art technology is the n -gram model [98], which estimates a probability distribution, $P(t_i|t_{i-n}^{i-1})$, where t_i is the current word considered and $t_{i-n}^{i-1} = (t_{i-n}, t_{i-n+1}, \dots, t_{i-1})$ is the observed previous $n - 1$ words. Since many n -grams are unobserved, smoothing techniques are often used to account for data sparsity, such as Katz smoothing [55] and Kneser-Ney smoothing [57].

2.2.4 Recognition

Recognition or decoding algorithms find the most likely sequence of words or phones given the temporally defined acoustic observations. The two most well known decoding algorithms for automatic speech recognition are Viterbi decoding [82] and A* search [93]. Since these two algorithms can be very time-consuming and computationally expensive, various simplification and pruning strategies have been utilized. For example, beam search [84] strategies eliminate a hypothesis if the associated probability is below the most probable hypothesis by a specified amount. Lexical trees [83] take advantage of the fact that most of the search is dedicated to the first few phones of a hypothesized word. Such methods have a disadvantage due to the inability to use language model scores prior to reaching the finish of a hypothesized word; therefore, look-ahead strategies [86] have been proposed.

Biography by William Ruhlmann

By the time Eric Clapton launched his solo career with the release of his self-titled debut album in mid-1970, he was long established as one of the world's major rock stars due to his group affiliations -- the Yardbirds, John Mayall's Bluesbreakers, Cream, and Blind Faith -- which had demonstrated his claim to being the best rock guitarist of his generation. That it took Clapton so long to go out on his own, however, was evidence of a degree of reticence unusual for one of his stature. And his debut album, though it spawned the Top 40 hit "After Midnight," was typical of his self-effacing approach: it was, in effect, an album by the group he had lately been featured in, Delaney & Bonnie & Friends... [Read More...](#)

Photo by Joseph Sia

Picture Browser
 < Previous Next >

Born
 Eric Patrick Clapp on **Mar 30, 1945** in **Ripley, England**

Years Active
 1910 20 30 40 50 60 70 80 90 2000

Genre	Styles
' Pop/Rock	' Blues-Rock
	' Pop/Rock
	' British Blues
	' Album Rock
	' Adult Contemporary
	' Hard Rock

Other Entries Influenced By

Watch music videos by this artist!

Figure 2.4: Eric Clapton spans multiple styles and genres.

2.3 Defining Musical Similarity

One fundamental problem in designing content-based classifiers is choosing an appropriate objective, i.e., measure of similarity. Early approaches assumed that songs belonging to a particular genre or artist could be considered “similar.” A single example demonstrates the inaccuracy of this assumption for artist-based similarity. Eric Clapton is a musical artist whose career spans multiple decades and styles, as demonstrated from his artist page on allmusic³ and presented in Figure 2.4. While he is given the rather ambiguous genre of pop/rock, he has played diverse styles such

³www.allmusic.com

as hard rock and adult contemporary. The view of a genre-based taxonomy for information retrieval purposes has its origins in an oral presentation of a study by Perrott and Gjerdingen [95], which demonstrated that people are more consistent in rating genres when given larger samples of songs. The effect was noticed until a ceiling of three seconds, and further listening durations did not improve consistency. However, a written version of the presentation remained unpublished until 2009 [42], a full decade after the results were first presented. Over time, this study became the most wrongly cited publication in MIR literature [7]. For instance, many authors have used this as reference for a performance ceiling; however, the authors now state that this is not their view, nor were their experiments designed to study such a hypothesis, as pointed out by [7] and by Perrot and Gjerdingen [42].

Defining a reliable ground-truth labeling scheme prior to the development of content-based algorithms is an obvious requirement. From a feature extraction point of view, it defines how to model the underlying signal, e.g., rhythmic features for ballroom dance classification. Further, the given task can impact classifier modeling; e.g., the semantics of musical chords may improve melodic content analysis.

It has been suggested that a “glass ceiling” [5] in performance exists for music similarity algorithms based on a timbral model, which has been strengthened by later studies [89]. However, in order to understand the true nature of this “glass ceiling” phenomenon, a reliable taxonomy is needed to resolve whether errors occur due to feature extraction and modeling or due to the subjective nature in defining music similarity. As an example of how a well-defined taxonomy can lead to improved results, one can imagine the task of language identification. If one noticed that there was a high confusion rate between Mandarin and Spanish, one could hypothesize that features that describe the tonal content in speech may improve performance because tonality changes the meaning of words in Mandarin.

It was noted by Pachet and Cazly [87] that existing genre-based taxonomies were derived by individual sources (e.g., record companies and retailers) for specific needs of the creators and lack consistency in both definitions and details. For example, while one person may be able to differentiate several genres of classical, such as Baroque, Romantic, etc., another might only know them by the parent genre, but he or she might recognize several more areas of rock music than a classical aficionado. Therefore, [87] develops a hierarchical taxonomy of genre by finding the factors that differentiate one group from its root genre. However, this can lead to common genres being ignored; e.g., rap is not used as a genre label because it is too broad. While this may benefit machine learning algorithms because of label consistency, the results require education on the part of the user for interpretability. Other solutions are based on user surveys [14] or games [126]; however, as noted by McKay and Fujinaga [76], general notions of similarity still possess the same problem of genre taxonomies, i.e., subjectiveness.

Recently, content-based recommendation algorithms have concentrated on *semantic tags*, which are short descriptions of a given object. Real-world examples can be seen on the social image website flickr⁴, the academic website citeULike⁵, and the social networking radio station, last.fm. Generally, semantic tags are unstructured and may be generated by any consumer of the given object; however, it is possible that restrictions may exist in both content (e.g., tag length) or authorship (e.g., open to the general public or closed to members or editors). In general, open systems are collaborative in nature, where it is assumed that the aggregated tagging results will present a more detailed and accurate view than could be derived from a single expert source, i.e., the “wisdom-of-the-crowds” phenomenon [120]. However, some systems restrict tagging to only a few individuals, such as the author of the content

⁴www.flickr.com

⁵www.citeulike.com

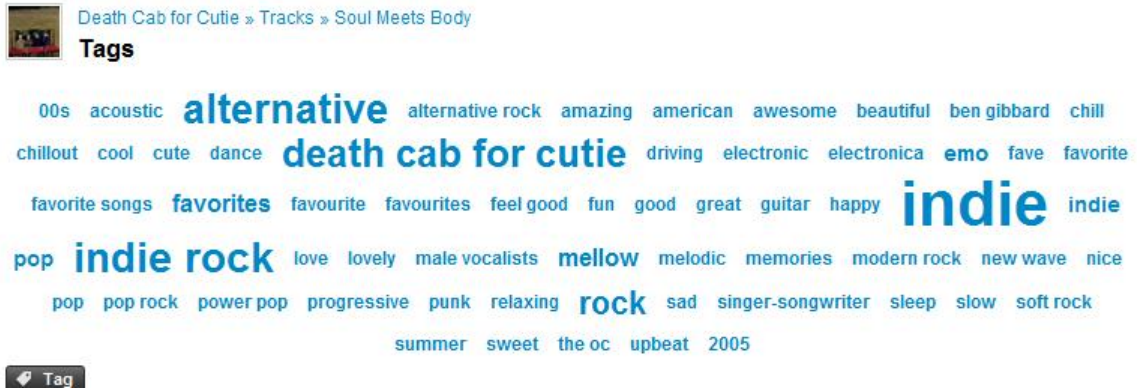


Figure 2.5: Tags for *Soul Meets Body* by Death Cab for Cutie from social networking site last.fm.



Figure 2.6: Tags for *Soul Meets Body* by Death Cab for Cutie from Pandora.

or an editor. An example from the unstructured, open last.fm radio station is seen in Figure 2.5, and an example from the structured, closed Pandora radio station is seen in Figure 2.6.

In terms of content-based recommendation algorithms, the goal is the same: to predict the presence of the hypothesized attribute given the raw, acoustic data. However, there are some significant qualitative differences between the two tag sets. Most obvious is the level of detail in terms of describing the acoustic content. The crowd-based tag set contains many words that describe high-level concepts such as genre,

but also contains information that is extraneous to the audio signal. Examples include personal opinions (e.g., “awesome”), time of year the song was released (e.g., “summer”), and a television show that used the song as background music (e.g., “the oc”). It is unlikely that one could design an acoustic-based classifier that would be highly accurate in estimating whether a song appeared in a television show or when a song was released. Further, tags like “awesome” are subjective in nature and may not be based on acoustic content. Meanwhile, the semantic tags from Pandora (Figure 2.6) contain information that has a higher chance of being extracted from the audio signal because the music qualities listed are less subjective and more quantifiable. Chapter 4 demonstrates how the temporal modeling achieved by the acoustic segment modeling approach is an improvement in content-based tag algorithms.

2.4 Ordinal Regression

While this dissertation presents an algorithm that leverages temporal modeling and tokenization to improve both genre classification and tag recommendation, an additional goal is to capture personalized similarity judgments. In this dissertation, a novel ordinal regression algorithm is proposed to predict user-specific notions of similarity using content-based analysis (see Chapter 6). The task of ordinal regression is described in this section and previous approaches are highlighted. Given a set of training data, $(\mathcal{X}, \mathcal{Y}) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^D$, $y_i \in \{1, 2, \dots, R\}$, D is the dimension of the data, and R is the number of classes considered, the goal is to find a function, $f(\mathbf{x}) \in \mathcal{Y}$. Further, it is assumed that a preference relation of $R \succ R - 1 \succ \dots \succ 1$ exists. The application of ordinal regression in Chapter 6 is to predict user ratings, e.g., “4 out of 5 stars.”

One such function that accomplishes this task is

$$f(\mathbf{x}) = \min_{r \in \{1, 2, \dots, R\}} \{r : \mathbf{w}^T \mathbf{x} - b_r < 0\}, \quad (2.5)$$

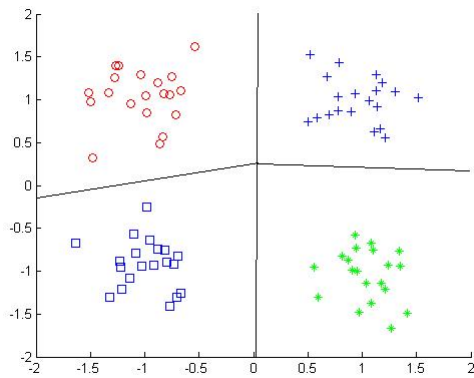
where \mathbf{w} is the weight vector common to all classes and b_r is the threshold for rank r .

The ordinal regression problem can be seen as a mixture of a multi-class classification problem and a regression problem, as shown in Figure 2.7. In Figure 2.7(a), the objective is to define a set of functions that classifies objects into a discrete set of categories. The regression problem pictured in Figure 2.7(b) demonstrates that the regression objective is to define a function that minimizes the error between the ordinal and the predicted value from the estimated function. However, in the ordinal regression problem shown in Figure 2.7(c), a discrete set of classes is constrained such that the decision boundaries are perpendicular to a linear function describing the dimension of preference. The decision boundaries between the ranks in (2.5) are parallel, but not necessarily evenly spaced. Generally, the quality of a ranking algorithm is given by the ability to minimize the average ranking loss:

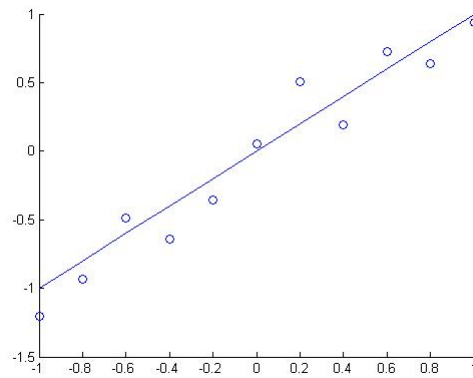
$$L_{AR}(\mathcal{X}; \mathbf{w}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|, \quad (2.6)$$

where \mathbf{b} is the vector $[b_1, \dots, b_{R-1}]$ and \hat{y}_i is the predicted rank found in (2.5).

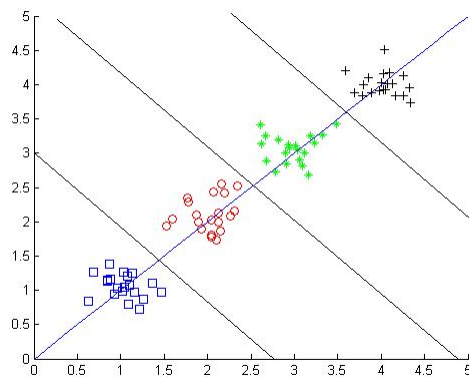
While it is possible to apply a regression formulation to the ordinal regression problem, such algorithms do not output a discrete class, thus are not a complete solution to the intended problem; i.e., that the output is constrained to be a discrete value. One of the first approaches to ordinal regression was based on the perceptron algorithm and called PRank [29]. The weight vector is coupled to all classes and updates to both the weight vector and the decision boundary are based on misclassified samples in the training data. A maximum-margin support vector machine (SVM) is given in [113], where explicit constraints are given such that no samples between neighboring classes are misclassified. A modification is given in [26] that adds constraints to maintain rank orders. The ordinal regression algorithm presented in this dissertation (see Chapter 6) minimizes (2.6), which is non-continuous and non-differentiable, by approximation with a sigmoid function. This approximation gives a continuous objective function, which can be minimized by gradient probabilistic descent [54].



(a) Multi-class classification



(b) Regression



(c) Ordinal regression or ranking

Figure 2.7: Differences between multi-class classification, regression, and ordinal regression.

CHAPTER III

ACOUSTIC SEGMENT MODELING FOR MUSIC INFORMATION RETRIEVAL

In this chapter, the acoustic segment modeling procedure for MIR is presented. As mentioned in Section 2.2, this approach is based on the success of speech recognition technologies. Specifically, [145] demonstrated that performance in automatic language identification improves when modeling phonetic content [46][124] versus modeling only the spectral content of the signal [81]. Current MIR approaches either ignore temporal information (e.g., GMM modeling of MFCCs) or model only short-time information (e.g., derivatives of MFCCs, texture windows, etc.). While the degree to which cognitive processes overlap in discriminating aspects of music versus aspects of speech is still an active research topic, there are similarities between different MIR tasks and ASR tasks from a signal modeling perspective [61]. Examples include the following:

1. Automatic speech recognition and automatic music transcription attempt to recognize the acoustic representation of a given message in the presence of a noisy channel.
2. Automatic speaker identification and automatic instrument identification recognize the sound production system.
3. Automatic language identification and automatic genre classification identify the rule set that most likely produced the given sound.

In general, the supervised ASR task is shown in Figure 3.1, where both acoustic and text representations of spoken utterances are given. The goal is to divide the

acoustic space into a meaningful set of models (e.g., words, phones, etc.) for later speech processing tasks. However, there are two sources of difficulty in adapting the supervised ASR task to MIR problems: defining an appropriate level of modeling capability (e.g., notes versus chords, see Section 2.1.2.3), and finding appropriate transcriptions.

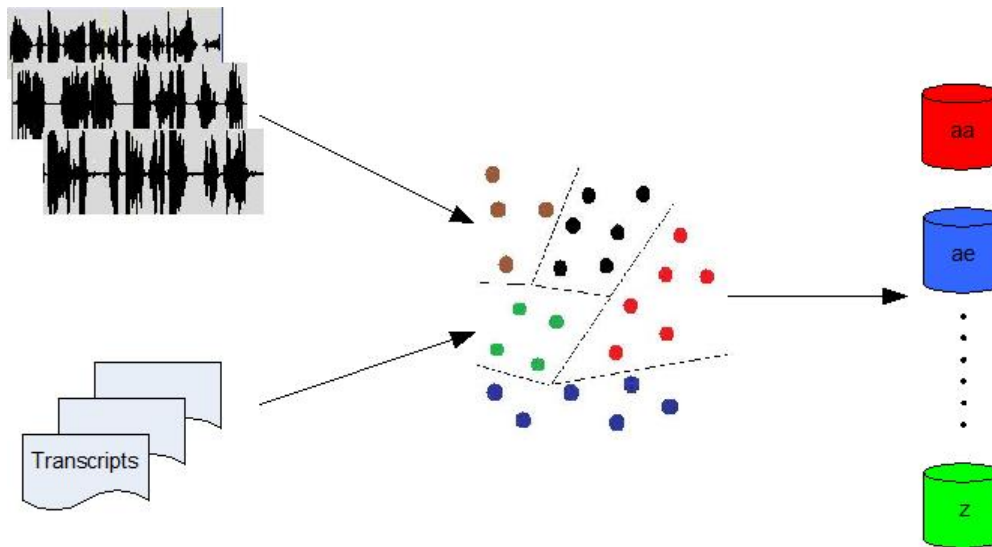


Figure 3.1: Modeling approach for automatic speech recognition.

As a solution, this chapter describes the acoustic segment modeling framework, which builds a vocabulary of acoustic tokens using an unsupervised process, as illustrated in Figure 3.2. Each of these acoustic tokens, called acoustic segment models (ASMs), is comparable to phonemes in speech because a given musical work is modeled as a temporal ordering of an intended ASM sequence. Further, each ASM that appears in a musical work is considered to be a noisy representation of the intended ASM; that is, each ASM is modeled probabilistically. Finally, by transcribing each musical work using the vocabulary set of ASMs, text-based retrieval algorithms can solve various MIR tasks. The block diagram is given in Figure 3.3 and the stages are described in the following sections.

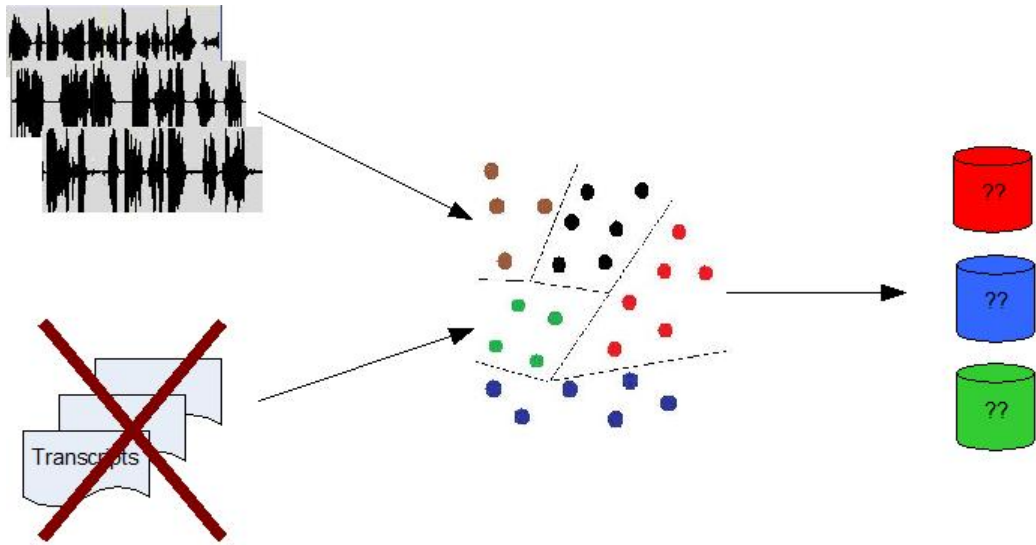


Figure 3.2: Modeling approach for music information retrieval.

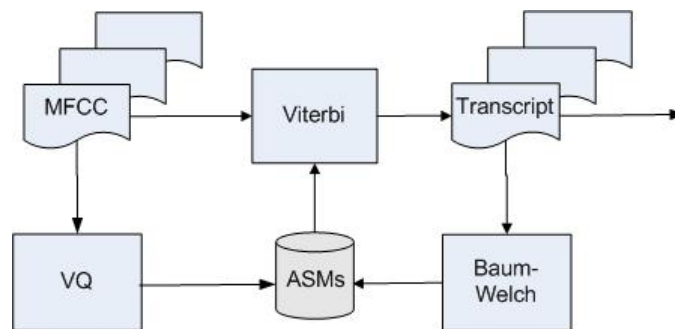


Figure 3.3: Acoustic segment modeling block diagram.

3.1 Feature Representation

As mentioned in Section 2.2.1, MFCCs are a common feature representation for speech [31] and have also been shown to be successful for music retrieval [70]. This chapter uses two MFCC representations. The first is solely used in the initialization of the ASMs (see Section 3.2) and describes the slowly changing spectral shape. The other type is more commonly used in ASR tasks and captures more spectral and temporal information.

The first step in the acoustic segment modeling procedure is to segment each song into regions that are fairly homogeneous. Ideally, the level of segmentation would compare with the base units of speech; i.e., phones. Each audio file is first divided into 25 ms, non-overlapping frames that are weighted by a Hamming window. The windows are chosen to be non-overlapping to decrease computation time, as this is the most time-consuming step in the algorithm. Note that later training stages will redefine boundary locations. Because the low-order MFCCs describe the slowly changing spectral shape [61], only the first eight MFCCs are extracted for the initial segmentation.

The second type of MFCCs is used in the remaining stages. While the frames are still 25 ms in duration, there is a half-frame overlap between successive frames. Each frame is then weighted with a Hamming window and 12 MFCCs (excluding the zeroth coefficient) and the log-energy are extracted. Each MFCC vector is concatenated by the first and second derivatives of the MFCC sequence to yield a 39-dimensional feature vector for each frame.

3.2 Initial Segmentation and Transcription

An initial set of ASMs is built by segmenting each acoustic music file, representing each segment by a feature vector, and clustering all segments across the training database. Each recording is segmented using a level-building, dynamic programming

algorithm [121]. It should be noted that this approach is time consuming and other segmentation schemes may produce better results. For example, if given the raw audio, beat segmentation algorithms provide a more musically intuitive segmentation. However, in later sections, data will only be available using MFCCs, and current beat and onset detection schemes use a different feature set; therefore, the maximum likelihood segmentation algorithm is used in all experiments.

The segmentation algorithm groups successive frames such that the following distortion function is minimized:

$$D(O, Q) = \sum_{q=1}^Q \sum_{t=s_{q-1}+1}^{s_q} d(\mathbf{o}_t, \mu_q), \quad (3.1)$$

where $O = (\mathbf{o}_1 \mathbf{o}_2 \dots \mathbf{o}_{|O|})$ is the sequence of $|O|$ observation vectors, μ_q is the centroid of the q^{th} segment that begins at frame $s_{q-1} + 1$ and ends at s_q ($s_0 = 0$), and $d(\mathbf{o}_t, \mu_q)$ is a distance metric between \mathbf{o}_t and μ_q , which is the Euclidean distance for this dissertation. The dynamic time-warping procedure described in [121] minimizes (3.1). An example of the most likely segmentation is shown in Figure 3.4 for a song from the GTZAN dataset [127]. The maximum likelihood approach results in sections that are maximally similar across a given segment and generally occur at note boundaries.

Next, each segment is summarized by the mean MFCC vector taken across all frames in a given segment. The k -means algorithm [72] groups the segment vectors into a set of N_{ASM} clusters, where N_{ASM} is the number of ASMs. By assigning each segment mean vector to the closest cluster, a transcript for each training file is created. An example of how these transcripts look is given in Figure 3.5. Each line in a file represents a cluster index. For example, “x123” is the first “word” in song 1, “x54” is the second, etc. Therefore, each song is represented by a sequence of symbols in the same way that a text document or speech transcription is a sequence of words.

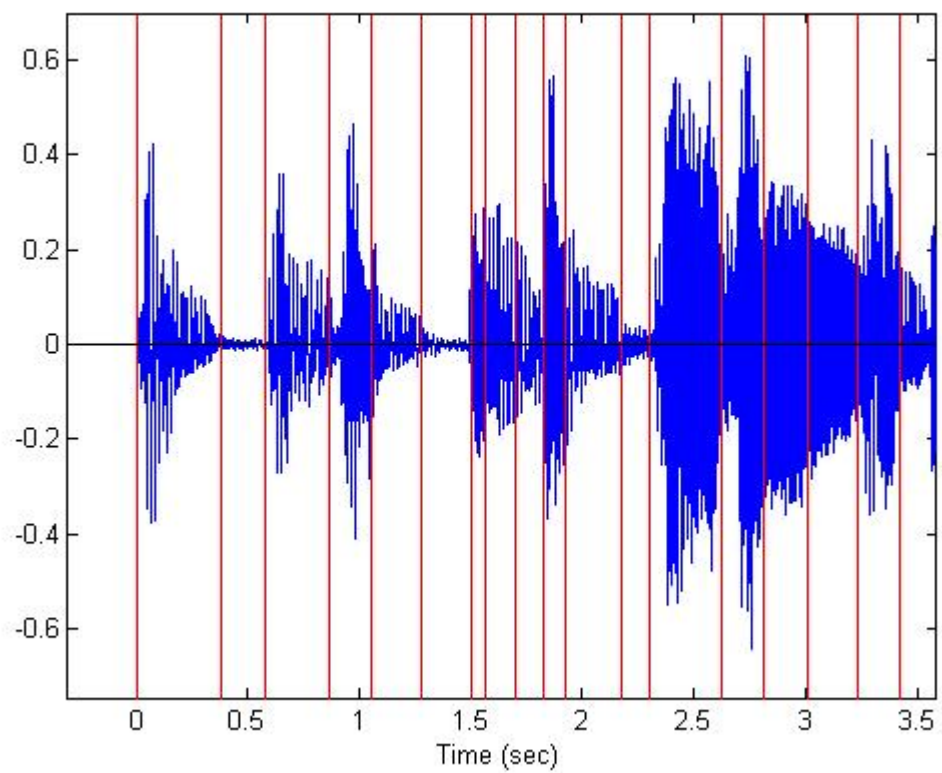


Figure 3.4: Initial segmentation of a song segment in the GTZAN dataset.

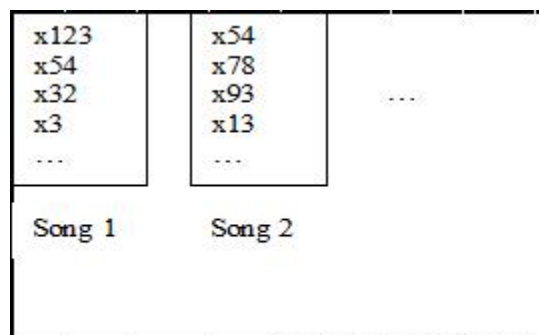


Figure 3.5: Example ASM transcript.

3.3 ASM/HMM Training

The transcripts obtained in the previous step provide a starting point for an iterative HMM training process, where each ASM is modeled with a left-to-right HMM. While the first eight MFCCs accomplish the task of finding the initial segmentation, it has been found that using higher-order coefficients, energy, derivative coefficients, and acceleration coefficients yields better results for modeling audio and speech content. Therefore, the 39-dimensional feature vector mentioned in Section 3.1 is used in this section.

First, Baum-Welch estimation [98] trains the set of HMMs using the training data. After Baum-Welch estimation, the HMMs are used to re-transcribe the set of training files into a new ASM sequence using Viterbi decoding [98]. These transcripts will be different from the original transcripts and are used to further train the HMMs in an iterative process between Baum-Welch estimation and Viterbi decoding. This process is repeated until an appropriate stopping condition is reached. At the end of the acoustic segment modeling training stage, a set of ASMs is produced that are used to tokenize a given song into a temporal ordering of ASMs by Viterbi decoding.

3.4 Experiments

In this section, the ability to use ASMs to describe the semantic information for genre recognition is explored. It has been suggested that music genre classification parallels the spoken language identification problem [61]. Specifically, with regard to music theory, a genre of music provides a probabilistic set of rules by which sounds are produced in terms of their spectral and temporal characteristics. For example, the basic 12-bar blues form specifies an ordering of I, IV, and V chords [129], which shows how the genre imposes syntactic constraints that influence transition probabilities between fundamental acoustic units, e.g., notes and chords. In addition, these fundamental units vary in both observational feature values and in duration. However,

it should be mentioned that these rules are not as restrictive as language rules and are often subjective. For example, music with an orchestral arrangement does not necessarily preclude the music from being in the rock genre. Despite the subjective nature of genre classification and the current debate over the utility of content-based genre recognition [4][76], automatic genre classification still remains one of the more active research topics at the yearly International Symposium for Music Information Retrieval (ISMIR) [44]. The proposed approach is to model the genre classification problem in the same fashion as a standard vector-based approach to topic identification in natural language processing. The approach is a two-stage process, shown in Figure 3.6.

3.4.1 Front-end Specifications

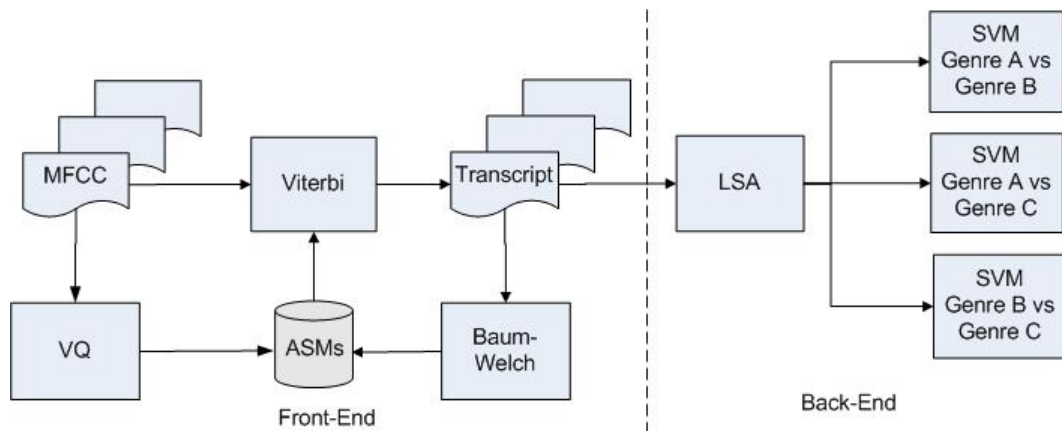


Figure 3.6: Block diagram for ASM genre classification.

The front-end is the acoustic segment modeling approach described previously in this chapter. Unless otherwise stated, the size of the ASM vocabulary is $N_{ASM} = 128$. Each ASM is modeled with a three-emitting state HMM, where each emitting state contains an eight-mixture GMM observation density. The iterations between Baum-Welch estimation and Viterbi decoding are performed until the percent accuracy defined in the HTK toolkit [142] does not change between iterations by more than

Table 3.1: Convergence of ASM procedure.

Iteration	1	2	3	4
Acc%	46.52	71.16	78.02	83.59

five percent. That is, for a given iteration, the transcripts used to estimate the parameters during Baum-Welch estimation serve as the reference transcripts and are compared to the new transcripts estimated from Viterbi decoding. If the percent accuracy changes by less than five percent, the process is terminated. In the HTK toolkit, the percent accuracy is found by finding the optimal string match between the new transcriptions and the reference transcripts. This optimal string match is found using dynamic programming and is used to find the percent accuracy:

$$Acc\% = \frac{N_L - D_E - S_E - I_E}{N_L} \times 100\%, \quad (3.2)$$

where N_L is the total number of labels in the reference transcriptions and S_E , D_E , and I_E are the number of substitution errors, deletion errors, and insertion errors in the optimal alignment, respectively. To demonstrate the convergence criteria, this measure is shown in Table 3.1 for the GTZAN dataset (see Section 3.4.3).

At the output of the front-end system is a set of ASMs, each modeled by an HMM. The models are used to decode both the training and testing files, so that each song is represented by an ASM transcription. Further, the ASMs can be viewed as words or even as an alphabet of an acoustic language. The sequence of ASMs can be seen as syntax, even if on a rough level.

While the academic debate about the overlap in cognitive processing of music and speech is still ongoing, there does seem to be some similarity. This can be seen with music theory, which dictates syntactical usage. As another example, a well-known phenomena in language processing is Zipf’s Law [144], which says if one ranks the terms in order of their frequency, $freq$, in a large corpus of any language then the

relationship between $freq$ and the rank is

$$freq \propto \frac{1}{rank}. \quad (3.3)$$

One surprising result when applied to the MIREX dataset is that Zipf's Law does not apply to the appearance of individual ASM counts (*unigrams*); however, by treating the consecutive ordering of two ASMs as a single term (*bigram*), this behavior is demonstrated, as shown Figure 3.7. One potential reason for this effect may be due to the fact that the information-carrying content of the signal is in the relationships between co-occurring sounds (both in time and frequency), rather than in the sounds themselves. For example, a single isolated word in speech carries information (e.g., subject, verb, etc.); however, a single isolated note contains very little information until it is placed in context by either notes co-occurring in a chord or in a melody. However, the authors caution that this is a hypothesis that needs further testing to confirm.

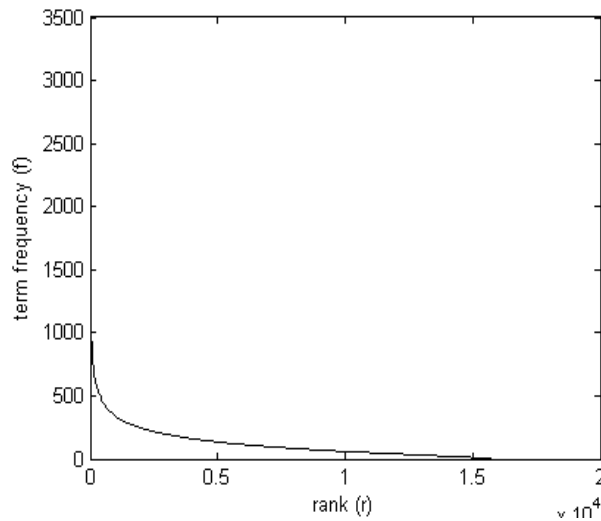


Figure 3.7: Bigram frequency versus ranking applied to the MIREX dataset.

3.4.2 Back-end Classifier

Following the front-end acoustic segment modeling procedure, each song is represented by a text document containing the temporal sequence of the ASMs most likely to have produced the acoustic realization of the song. The back-end system is motivated by a common vector-based strategy for topic categorization. First, latent semantic analysis (LSA) [74] produces a vector of weighted term counts for each musical piece. Next, the final decision is given by a majority vote of binary classifiers, where each classifier is a SVM [131] that distinguishes between two genres. Details of each step are given in the subsections below.

3.4.2.1 Latent Semantic Analysis

Let $\mathcal{T} = (t_1, t_2, \dots, t_{|\mathcal{T}|})$ represent a given vocabulary set, with $|\mathcal{T}|$ items. The j^{th} document is converted into a vector of unigram counts, $c_{.j}$, where the i^{th} element in the vector is

$$c_{i,j} = \sum_{m=1}^{M_j} \delta(\omega_m, t_i), \quad (3.4)$$

where the j^{th} document is a word sequence, $\omega_1, \omega_2, \dots, \omega_{M_j}$, of length M_j , $\omega_m \in \mathcal{T}$, and $\delta(x, y)$ is one if the two arguments are equal and zero otherwise. For this dissertation, $c_{.j}$ in (3.4) is extended to include bigram counts. To maintain clarity, the words unigram and bigram will be used when specifically referring to a particular type of n -gram. When referring to both or either unigram and bigram, *term* will be used.

In text retrieval, certain terms, such as *a*, *the*, *on a*, etc., are not very informative in summarizing the content of a document while other terms are very informative, such as *NASA*, *cepstrum*, *Georgia Tech*, etc. Therefore, term counts are often weighted by the product of two features: the term frequency and the inverse document frequency. The former represents the probability of observing t_i in the current document and is given by

$$tf_{i,j} = \frac{c_{i,j}}{\sum_{m=1}^M c_{m,j}}, \quad (3.5)$$

where M is the number of terms considered (e.g., $M = |\mathcal{T}| + |\mathcal{T}|^2$ when unigrams and bigrams are considered). The latter penalizes terms that appear in more documents and is given by

$$idf_i = \log \frac{N}{|j : w_i \in j|}, \quad (3.6)$$

where N is the number of song documents in the training set. The product of these terms provides the weighted term count:

$$a_{i,j} = tf_{i,j} \times idf_i. \quad (3.7)$$

Concatenating the N document vectors forms an $M \times N$ term-document matrix, A , which is both large and sparse. For example, with a vocabulary size of 128, each document vector has a length of $M = 128 \times 128 + 128 = 16512$ when both unigrams and bigrams are considered. Sparsity arises because many unigrams and bigrams appear in only a few documents; therefore, singular value composition (SVD), which is similar to eigenvalue decomposition [119], is utilized. SVD decomposes the term-document matrix as

$$A = U \Sigma V^T, \quad (3.8)$$

where U is $M \times \rho$, Σ is $\rho \times \rho$, V is $N \times \rho$, and ρ is the rank of the original matrix, A . The left-singular matrix, U , and the right-singular matrix, V , are orthonormal and represent the term and document space, respectively. The matrix Σ is a diagonal matrix of singular values. By keeping the ρ_0 ($\rho_0 < \rho$) largest singular values, the term-document matrix can be converted into a lower-dimensional “concept” space [11]. Unless otherwise stated, ρ_0 is chosen such that

$$\frac{\sum_{i=1}^{\rho_0} \lambda_i^2}{\sum_{i=1}^{\rho} \lambda_i^2} \leq 0.9, \quad (3.9)$$

where $\lambda_1^2 \geq \lambda_2^2 \geq \dots \geq \lambda_\rho^2$ are the singular values in Σ .

The reduced document matrix, V_{ρ_0} , formed by retaining the vectors in V corresponding to the largest singular values, serves as the training data for the final

classifier. During testing, a query song is converted into a weighted term vector of ASM counts, \mathbf{q} , and projected it into the “concept space” by

$$\mathbf{q}_{\rho_0} = \mathbf{q}^T U_{\rho_0} \Sigma_{\rho_0}^{-1}, \quad (3.10)$$

where U_{ρ_0} and Σ_{ρ_0} are constructed in the same fashion as V_{ρ_0} .

3.4.2.2 Support Vector Machines

SVMs [131] are binary classifiers motivated by statistical learning theory. Given a set of training data pair, $(\mathcal{X}, \mathcal{Y}) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \{-1, +1\}$, the goal is to find a function that can classify future examples with as small an error as possible. Generally, a linear function is used:

$$f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b, \quad (3.11)$$

where \mathbf{w} and b are the decision hyperplane and bias parameter, respectively. Note that for any point, \mathbf{x}_0 , on the line defined by (3.11), $f(\mathbf{x}_0) = 0$. In addition, the vector \mathbf{w} is normal to the separating hyperplane; therefore, $f(\mathbf{x})$ is proportional to the distance between the line and the point \mathbf{x} . To see this, let \mathbf{x}_0 be the closest point on the hyperplane to \mathbf{x} . Then, $\mathbf{w}^T (\mathbf{x} - \mathbf{x}_0) = \mathbf{w}^T \mathbf{x} - b = f(\mathbf{x})$. A classification rule is induced by noting the sign of (3.11), as shown in Figure 3.8.

The classification problem in Figure 3.8 is a linearly separable two-class problem. There are an infinite number of solutions that separate the two classes perfectly; however, not all solutions are equally valuable. For example, if there is higher risk in misidentifying the “x” class versus the “square” class or if it is known that examples in the “x” class are noisier than the “square” class, it is ideal to give more margin to the “x” class and to place the dividing line closer to the “square” class. However, in the absence of such information, it would be ideal to maximize the minimum margin to either class, which is obtained by finding the hyperplane that is equidistant to both classes.

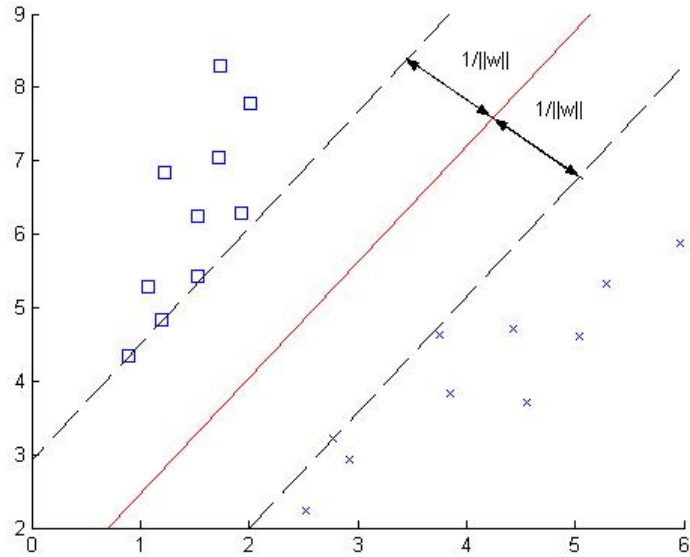


Figure 3.8: Support vector machine for a 2-class problem. See text for details.

This objective can be written as

$$\begin{aligned} \max_{\mathbf{w}, b, \|\mathbf{w}\|=1} \quad & C \\ \text{such that} \quad & y_i (x_i^T \mathbf{w} + b) \geq C, \quad i = 1, \dots, N, \end{aligned} \quad (3.12)$$

where C is the margin between the hyperplane and the closest training point. For any \mathbf{w} and b satisfying the constraints in (3.12), any positive scaled multiple will also satisfy them; therefore, by setting $\|\mathbf{w}\| = 1/C$, (3.12) is equivalent to

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{such that} \quad & y_i (x_i^T \mathbf{w} + b) \geq 1, \quad i = 1, \dots, N. \end{aligned} \quad (3.13)$$

Additionally, for cases where the data are not perfectly separable, slack variables may be added to (3.13). In either case, the solution may be found by transforming (3.13) into its Lagrangian dual and using standard optimization techniques, e.g., [80].

Although the distance to the decision hyperplane can be seen as a measure of confidence, SVMs are not necessarily calibrated; therefore, direct comparison between scores of two different SVMs is sub-optimal. Therefore, to extend the binary SVM

classifier to the multi-class problem, a one-versus-one voting strategy is used [36]. Given $|\mathbb{C}|$ genres, a total of $|\mathbb{C}|(|\mathbb{C}| - 1) / 2$ SVMs are constructed, where each SVM uses a different genre pair. For example, if only rock, blues, and rap are considered, then an SVM is constructed using rock as the positive class and blues as the negative class. Another SVM is constructed using rock as the positive class and rap as the negative class. The final SVM uses blues as the positive class and rap as the negative class. A test sample is presented to all three SVMs and a “vote” is given by each SVM. The final decision is given to the genre that collects the most votes. This dissertation implements the SVMs using *SVM^{light}* [52].

3.4.3 Database and Evaluation

There are two datasets used to test the ability of ASMs to categorize songs by music genre. The first is a set of songs from the license-free music website, Magnatune¹ and used at the 2004 Audio Description Contest². The dataset contains the following genres, with the numbers of songs for each genre given in parenthesis: classical (109), electronic (115), jazz and blues (53), ambient (50), and rock/pop (92). Direct comparison of the results using this dataset is difficult because the evaluation reported at the 2004 Audio Description Contest is based on a withheld test set that has not been released. Also, the dataset is relatively small and distributed unevenly between genres; therefore, for the purposes of training the HMMs, the RWC [43] and Dortmund [48] databases are added. However, because different datasets will have genre labels based on different criteria [4], only the Magnatune dataset is utilized for training and testing the final genre classifiers. To increase the number of training vectors, each song is divided into 30-second segments. Similarly, test songs are also divided to account for underrepresented genres. An artist filter [89] ensures that no artist overlaps between the training and testing set.

¹www.magnatune.com

²http://ismir2004.ismir.net/ISMIR_Contest.html

The second dataset, called the GTZAN dataset [127], contains a more balanced set, with 100 30-second song segments for each of ten genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. Comparisons to other methods are possible with this dataset since it is publicly available. For this dataset, a ten-fold evaluation is performed ten times. That is, the dataset is divided into ten sets randomly. The experiment is conducted by using one of the sets for testing and the remaining nine sets for training. The experiment is repeated nine more times, but using each of the other nine sets for testing. The entire procedure is repeated nine more times, but with a different random split of the data.

The performance is measured in terms of accuracy, which is the percentage of correct guesses and is the measure reported at the MIREX competition. In addition, precision and recall rates are given, which are, respectively,

$$recall = \frac{tp}{tp + fn} \tag{3.14}$$

and

$$precision = \frac{tp}{tp + fp}, \tag{3.15}$$

where tp , fp , and fn are the number of true positives, false positives, and false negatives, respectively.

3.4.4 Results

As stated in Section 3.3, the training of HMMs is an iterative process between updating the ASMs and creating new transcripts using the updated ASMs. To view how the testing data responds to this process, the genre accuracy percentage for the first four iteration rates on the MIREX dataset is shown in Table 3.2. The accuracy rates on the test set increase each time a new set of transcripts for the training data is created and the HMMs are retrained with the new transcripts. There does appear to be an asymptotic value close to 73%, which is consistent with previous solutions to this problem. Interestingly, this finding does suggest that the “glass ceiling” in

Table 3.2: Accuracy versus iteration number.

Iteration	1	2	3	4
Acc%	67.87	69.32	72.14	72.86

Table 3.3: Confusion matrix for the Magnatunes dataset, given as counts. The last row and column are the precision and recall percentages, respectively.

Genre	Classical	Electronic	Rock	Jazz/Blues	Ambient	Recall
Classical	26	0	1	1	2	86.7
Electronic	0	19	9	0	2	63.3
Rock	0	5	24	0	1	80.0
Jazz/Blues	1	2	5	12	1	57.1
Ambient	1	4	2	1	21	72.4
Precision	92.9	63.3	58.5	85.7	77.8	

performance noted in [5] is not strictly due to a lack of temporal modeling, but is due to other factors. This is investigated further in Chapter 5.

The final confusion matrix is displayed in Table 3.3, where the rows represent the ground truth and the columns represent how the algorithm classified the test songs. Most errors occur in just one other class and can be explained by the fact that many songs are not necessarily strictly jazz, strictly electronic, etc. For instance, some of the files in the Magnatunes corpus are described as “electronic rock with a pop edge.” This may indicate that many of the proposed genre classification schemes need to be extended to allow for multi-topic categorization. Additionally, heuristics based on perception and cognition may help in discrimination.

One issue with the MIREX dataset is the inability to compare against previous studies. A more standard comparison can be made on the GTZAN dataset [127], mentioned in Section 3.4.3. The acoustic segment modeling procedure is compared to previously published results in Table 3.4. It should be noted that Bergstra et al. [15] use boosting, which can be easily incorporated to extend the back-end classifier. In addition, Lidy et al. [68] uses symbolic information in addition to features extracted from the audio. The approach by Tzanetakis and Cook [127] extracts many different

Table 3.4: Comparison of genre classification accuracy on the GTZAN dataset between the ASM procedure and selected published results.

Reference	Accuracy
Bergstra et al. [15]	82.50%
Lidy et al. [68]	76.80%
ASM	74.80%
Tzanetakis et al. [68]	61.00%

Table 3.5: Genre classification accuracy when using an ASM vocabulary of 64 and 128.

64 ASMs	128 ASMs
70.90%	74.80%

rhythmic, pitch, and timbral features, including MFCCs, as inputs into a GMM. This is considered the baseline classifier since HMMs are the natural temporal extension to GMMs.

An important parameter is the number of ASMs in the vocabulary. If the number of ASMs is too small, there will not be enough acoustic coverage; however, too many ASMs will lead to a large dimensionality and require more training data and computation time. A comparison of different ASM vocabulary sizes for the first training/testing fold is shown in Table 3.5. It is seen that the accuracy increases when the number of ASMs is increased; however, increasing the number of ASMs to 256 was not computationally feasible.

3.5 Summary

This chapter presents the acoustic segment modeling framework for MIR. Previous approaches to MIR have ignored temporal context information. Such “bag-of-frames” models assume that short segments of a given song are observations from an independent and identically distributed process. However, songs often contain multiple notes and instruments; therefore, these different sounds are not produced by identically distributed sources. In addition, there are different levels of musical syntax, such as

short-term syntax (e.g., within note structure), mid-term syntax (e.g., melody), and even long-term syntax (e.g., song structure). Therefore, the different sounds within in a song are actually very dependent on one another.

Similar insights in the language identification task resulted in the phone recognition followed by language modeling (PRLM) approach [145], which has been shown to produce higher classification accuracies than spectral approaches. This is due to modeling temporal structure in the short-term (i.e., HMMs model each phone) and mid-term to long-term (i.e., n -grams). However, applying a similar approach to music is not a straight forward problem. Generally, ASR approaches assume a single source (i.e., speaker) or assume that co-occurring speakers are noise and interfere with the speaker of interest, i.e., the “cocktail party” problem [21]. Therefore, the dominant modeling paradigm (i.e., HMMs) is suited for this task without source separation techniques. Approaches to this problem either adapt existing HMMs trained in one background to HMMs suitable to the test condition (e.g., MLLR [66], MAP [41], etc.), modify the signal (e.g., spectral subtraction [19], cepstral mean subtraction [2], etc.), or modify the signal features (e.g., SPLICE [34], fMLLR [39], etc.) prior to modeling. However, such technologies are not suitable for music because co-occurring acoustic events are important to the understanding of the song. For example, co-occurring notes build chords and co-occurring instruments present a soundscape to the listener. Since source separation and polyphonic pitch estimation are unsolved problems, they are too error-prone as a front-end system for classification schemes.

Therefore, the acoustic segment modeling framework provides a transcription of music elements, which function in a similar fashion as phonemes for ASR. Further, because it is unclear as to what constitutes an appropriate model structure for music a priori (e.g., notes, chords, instrument mixtures, etc.), the ASM transcriptions are built using an unsupervised process. First, a given set of songs is segmented by a maximum likelihood approach, and each segment is modeled by a feature vector. A

codebook of ASM vectors is built by vector quantizing all the segments from all the songs into an appropriate number of ASMs. To model the short-time structure of sounds, the unsupervised transcriptions are used in the standard automatic speech recognition modeling framework; i.e., Baum-Welch estimation is used to model each ASM with an HMM. Viterbi decoding generates new unsupervised transcripts using the set of HMMs. The new transcriptions are used to update the models using Baum-Welch to start in an iterative process between Baum-Welch estimation and Viterbi decoding. Finally, typical text-based approaches are then utilized on the final unsupervised transcripts to make a classification decision.

For this chapter, the task of music genre classification was chosen. Results indicate that this approach is competitive with current methods and better than a similar approach [127]. While it was observed that the acoustic segment modeling framework benefits from the iterative training process and that increasing the number of ASMs produced better accuracy, the ability of ASMs to highlight temporal structure is further investigated in the next chapter.

CHAPTER IV

MUSIC STRUCTURE DETECTION

The previous chapter introduced the acoustic segment modeling procedure for MIR. It is shown to be a competitive approach to music classification tasks, such as genre recognition. To understand how the acoustic segment modeling procedure improves temporal structure detection, this chapter examines the acoustic segment modeling procedure on two tasks: temporal tag identification and musical chord recognition.

4.1 Temporal Tag Identification

As noted in Section 2.3, researchers have been shifting their focus towards content-based tag annotation due to the success of flickr, last.fm, and other social tagging sites for multimedia. For a historic overview on the transition from automatic genre detection to music tag annotation, please refer to [7]. The next few subsections describe an experiment demonstrating the ability of the acoustic segment modeling procedure to capture temporal tag information better than current approaches. First, a comparison of current approaches to using semantic tags for music retrieval is given. Next, the proposed acoustic segment modeling procedure for tag identification is detailed. A baseline “bag-of-frames” algorithm [125], which performed well at the 2008 MIREX Tag Classification Contest¹, is then presented. The database and evaluation metrics are then discussed. Finally, experimental results are presented.

4.1.1 Semantic Multimedia Tags

As described in Section 2.3, semantic tags are essentially keywords applied to a given object, but may describe other factors besides the content of the object. For example,

¹www.music-ir.org/mirex/2008/index.php/Audio.Tag.Classification.Results

semantic tags can highlight under what settings a user enjoys listening to particular song. Systems use semantic tags as features to recommend items that are novel to a user. Generally, such systems rely on techniques similar to collaborative filtering. Each tag can be seen as a third-order tensor $\langle user, item, tag \rangle$ [24]. While approaches exist that directly use the third-order tensor model [122], most approaches unfold the tensor based on the application [24]. For example, a two-dimensional matrix can be created where each row is a user, each column is an item, and each entry indicates whether the user has tagged an item. Alternatively, the two-dimensional matrix can have each row represent a tag, and each column represent an item, and each entry is the salience of the tag in the item. Note that this approach is similar to a “bag-of-words” vector-based model for document classification and retrieval [74].

Since tagging is similar to collaborative filtering, many of the same problems exist, such as the cold-start problem and the popularity problem (see Section 2.1.1). In addition, other problems exist for tag-based systems that do not exist to the same degree in collaborative-filtering systems. For example, hacking is common in many state-of-the-art tagging systems, where users maliciously try to direct query searches by applying false tags or by inflating tag scores by applying a tag multiple times [60]. Another problem is synonymy and misspellings, which can be approached by the similar problem seen in text retrieval [74]. However, another problem is polysemy, where the same word may have a multiple meanings. While approaches from natural language processing attack polysemy by using contextual knowledge, such as a thesaurus like WordNet [79], contextual cues are absent in tags because they are only a few words in length [63]. In addition, the speed at which a word changes meaning can be intensified due to the collaborative nature that often leads to a “folksonomy” [130]. For example, a malicious user may label a pop artist with the tag *death metal*. While this an example of hacking, it is also another definition for *death metal*, i.e., sarcasm meant to demonstrate contempt for *bubblegum* music.

Many of these problems are potentially solved by content-based tag identification algorithms. While the most obvious issues solved by content-based systems are the cold-start and popularity problems, other issues such as polysemy can be solved by analyzing the differences in acoustic content when the tag is applied. In fact, researchers had investigated the link between acoustic representations of music and their textual description prior to many social tagging sites. For example, [136] estimates a set of semantic basis functions to maximize the semantic meaning of words based on musical features. Slaney [116] models the connection between anchor points in the acoustic space and semantic audio descriptions in a hierarchical multinomial clustering model. Unlike these early approaches, current tag identification algorithms benefit from the fact that tags are more explicit than freely flowing text; therefore, it is possible to directly model the tags acoustically. In [16], a boosting algorithm is used on several acoustic features to predict whether a song should be labeled with a given tag. The system proposed by Turnbull, *et. al.* [125] serves as the baseline algorithm for this section and models each song with a GMM. A tag-level GMM is then trained using the song-level GMMs in a mixture-of-hierarchies algorithm. More details of this approach are given in Section 4.1.3.

4.1.2 Acoustic Segment Modeling for Tag Identification

The front-end stage in the proposed approach is the acoustic segment modeling procedure described in Chapter 3; however, it would take too long to segment every song in the training database using the maximum likelihood segmentation procedure described in Section 3.2. Therefore, a small set of songs that does not overlap with either the training or testing datasets in terms of artists or song titles is used to bootstrap an initial set of ASMs. That is, these initial seed songs are segmented using the maximum likelihood procedure and the segments are vector quantized using the k -means algorithm [72] to produce a codebook of initial ASMs. Initial transcripts are

created by noting the cluster assignments, and Baum-Welch estimation builds a set of HMMs (one for each ASM). Next, Viterbi decoding produces a new set of transcripts for the full set of training files, which are used for the remaining iterations of Baum-Welch estimation and Viterbi decoding. Each ASM is modeled with a three-emitting state HMM and each state has a 16-mixture GMM as the observation density.

After the final ASM transcripts have been obtained, LSA is performed (see Section 3.4.2.1). A total of 128 unigram ASMs plus their bigrams is considered to give a 16,512-dimensional vector of entropy-weighted term counts for each song. SVD reduces the dimensionality and sparseness of the data by retaining only the top 250 singular values, which was experimentally determined by reserving a part of the training set for cross-validation. A binary SVM is created for each individual tag, and the decision of each SVM is independent of the other tag SVMs. Note that the SVM linear function in (3.11) provides an output score that is the distance from the given sample to the separating hyperplane. Because the classification rule is from the sign of (3.11), the magnitude can be used as a measure of confidence [111]. The distances from the hyperplane are compared to a threshold to give a decision on whether a tag is present. However, note that SVM scores are not compared across SVMs because SVMs distances are not calibrated, which makes attempts to compare scores between SVMs difficult [111].

4.1.3 Baseline Tag Identification Algorithm: Mixture-of-Hierarchies

The “bag-of-frames” classifier in [125] serves as the baseline to test whether temporal modeling improves performance for tag annotation and retrieval. This algorithm uses a mixture-of-hierarchies [132] approach, which is modified in [125] to incorporate a salience weight, as shown in Figure 4.1. First, each song is segmented into small overlapping windows (approximately 25 ms in duration) and a feature vector is extracted for each time window. A multi-dimensional, eight-mixture GMM is estimated for

each song using the extracted feature vectors. Next, a 16-mixture GMM is estimated for each tag using the song-level GMMs. The amount that each song contributes to a tag is determined by a salience weight, which is determined in [125] by a group of listeners. Because the dataset used in this section (see Section 4.1.4) only contains information pertaining to the presence of an attribute, the weights are set to zero (tag absent) or one (tag present).

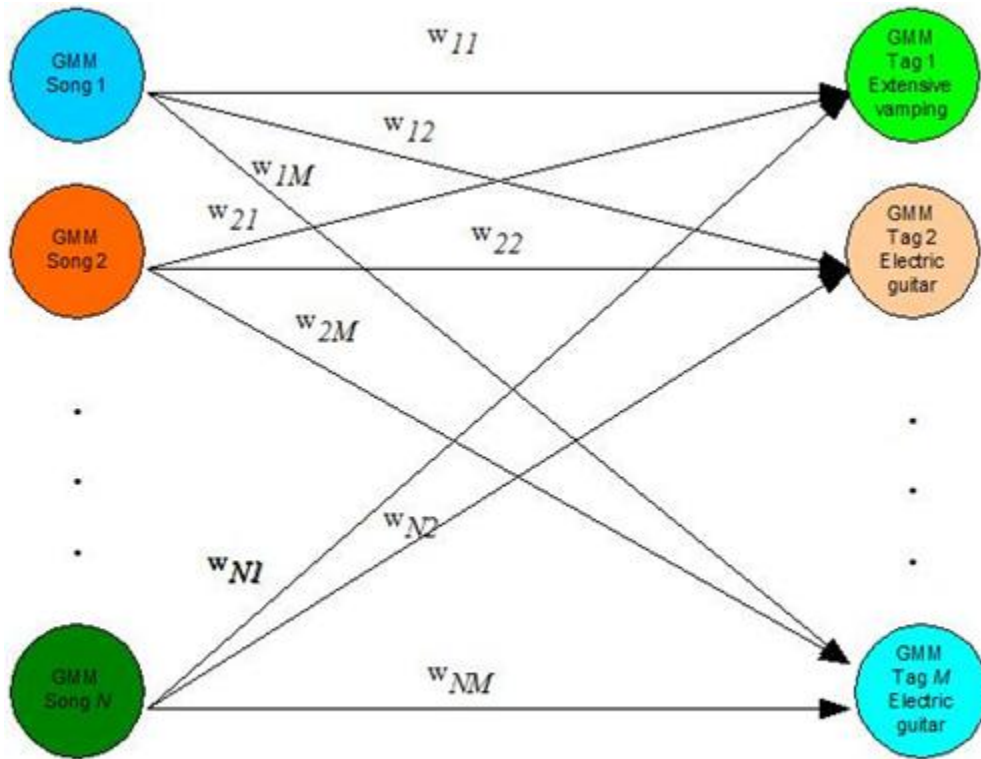


Figure 4.1: Diagram of mixture-of-hierarchies algorithm used in [125]. Weights between a given song and a given tag represent the salience of the tag in the song.

While [125] returned the top 10 tags for each song, a binary decision is needed for each tag in this section; therefore, an anti-tag model is created for each tag. The anti-tag model is similar to the idea of a cohort model for speaker verification [107]. The anti-tag model is created in the same fashion as the tag models, but with the polarity of the weights flipped. In reality, this may be closer to a universal model [107]

because Pandora lists only the most salient tags; therefore, an omission in the tag list does not mean the tag is absent. A log-likelihood ratio (LLR) is applied between the tag model and the anti-model and compared to a threshold for the decision.

4.1.4 Results

The dataset used in this section is the songs from the USPop dataset [14] for which a Pandora attribute list was found. Pandora was chosen because the attributes are less subjective than other websites and are aimed at describing the acoustic content of the signal. Of the 8,764 songs in the USPop dataset, 3,108 song descriptions were found. A tenfold evaluation is used, where the database is split into 10 equally sized, non-overlapping sections. An artist filter [89] ensures no artist appears in more than one split. The experiment is repeated ten times, where one of the ten splits is the test data and the remaining nine are used as training data. To ensure that enough test examples appear for each attribute, only attributes with more than 500 occurrences are considered, resulting in a list of 19 attributes. An advantage of such a small tag set is that qualitative comparisons are more easily addressed. Prior to the experiment, each of the 19 attributes was identified as either containing global information, temporal information, or both.

Performance is measured in terms of equal error rate (EER) for annotation and mean average precision for retrieval. EER is the point at which the false acceptance rate and false rejection rate are equal. Mean average precision gives the precision at each recalled document. For example, if the system returns the ordered results of [hit, miss, hit], then the mean average precision is [1,0.5,0.67] and is 0.5 at level two. The McNemar’s test is used to detect statistical significance and is a non-parametric statistical test that determines whether two classifiers are significantly different. It has been shown to have a low Type I error compared to other statistical tests [35].

Two tasks are performed in this section: annotation and retrieval, which mirrors that found in [125]. For annotation, [125] listed the top ten tags for each song; however, this is a little heuristic in nature because some songs may have more or less relevant tags. Further, it is necessary to develop a “best-case” to compare results. Therefore, this paper extends the approach in [125] to force a decision for each tag (see Section 4.1.3) by building an anti-tag model and comparing the two models with a LLR test. Similarly, the SVM scores in the acoustic segment modeling approach are compared to a threshold. For retrieval, the LLR test scores and SVM scores are sorted, and songs at the top of the list for a given tag are returned.

4.1.4.1 Annotation Results

The proposed acoustic segment modeling approach performs better for annotation in terms of EER for 15 of the 18 tags, as shown in Table 4.1. However, by analyzing the results based on the temporal characteristics, more interesting results are obtained. The table is organized by ranking the differences in EER by the t -statistic [92], so the improvement of the acoustic segment modeling approach over the baseline is largest for the tags at the top of Table 4.1. With the exception of *acoustic & electric instrumentation*, all tags appearing in the top half of the table contain temporal aspects, showing the ability of the ASMs to capture temporal information. Only four tags failed the McNemar’s test at a significance level of 0.05, which shows that for most tags, the best performance is not due to randomness in the training and testing sets.

Another interesting result is that the best performing tags under the acoustic segment modeling approach largely contain aspects of timbre, whether global or temporal. Examples include *electric rock instrumentation* (EER = 35.20) and *acoustic rhythm guitars* (EER = 28.74). The worst performing tags contain aspects of melody (*repetitive melodic phrasing* (EER = 44.59) and *melodic songwriting* (EER 46.35))

Table 4.1: Results for each tag in terms of EER for the proposed (Prop) and baseline (Base) approaches. Tags are labeled as either temporally-based, globally-based, or both, as indicated in the parenthesis. Bold face indicates McNemar statistical significance.

Tag/Attribute	ASM	Baseline
major key tonality (Temporal/Global)	34.54	42.76
electric guitar riffs (Temporal)	40.78	54.30
minor key tonality (Temporal/Global)	40.20	48.87
acoustic & electric instrumentation (Global)	39.23	48.24
acoustic rhythm guitars (Temporal/Global)	28.74	33.25
vocal harmonies (Temporal/Global)	41.74	47.77
extensive vamping (Temporal)	42.24	45.56
focus on studio production (Temporal/Global)	39.24	43.53
subtle use of vocal harmony (Temporal/Global)	41.15	43.68
mild rhythmic syncopation (Temporal)	46.64	49.16
a vocal-centric aesthetic (Global)	43.65	44.69
a dynamic male vocalist (Global)	43.65	44.69
hard rock roots (Temporal/Global)	19.13	19.44
melodic songwriting (Temporal)	46.35	46.67
electric rock instrumentation (Global)	35.20	34.70
acoustic rhythm piano (Temporal/Global)	37.58	35.79
repetitive melodic phrasing (Temporal)	44.59	41.92

and rhythm (*mild rhythmic syncopation* (EER = 46.64)). The authors conjecture that the poor performance in attributes describing melody is due to the choice in features, i.e., MFCCs. Features designed to model pitch, such as pitch class profiles [38] (see Section 4.2.1) should lead to superior performance. In addition, rhythm is largely affected by the granularity of the segmentation algorithm, which is the maximum likelihood segmentation algorithm described in Section 3.2.

4.1.4.2 Retrieval Results

An important application of semantic tags is retrieval, where one searches by semantic tags or keywords and is returned a list of relevant songs. To measure retrieval for the acoustic segment modeling and baseline approaches, the results for each tag are sorted by the LLR and SVM scores for the baseline approach and the acoustic segment

Table 4.2: Retrieval mean average precision for the acoustic segment modeling and baseline approaches.

Level	Proposed	Baseline
5	0.4477	0.3313
10	0.4249	0.3321
15	0.409	0.3277

modeling approach, respectively. Table 4.2 demonstrates that the acoustic segment modeling approach performs better in terms of mean average precision at the levels considered.

4.1.4.3 Temporal Analysis

To understand how the acoustic segment modeling procedure is able to capture semantic information, Figure 4.2 shows two parts of the same song that contain similar electric guitar riffs. The lower waveform is a solo, with a single, clean electric guitar and the upper waveform has an additional high-hat and finger snap. The acoustic segment modeling procedure finds an underlying *timbral melody* with the sequence (x33, x70, x29, x119, x33); however, the upper waveform also has *timbral embellishments*. Specifically, the finger snap at 41.9 seconds causes the insertion of the sequence (x29, x94) between x119 and x33. Further, the high-hat hit at 42.3 seconds causes the last x33 to repeat. The acoustic segment modeling procedure is able to identify that two musical pieces have locally similar characteristics, even when additional instruments are added. Note that a GMM would only detect that these sounds occurred, but not their ordering

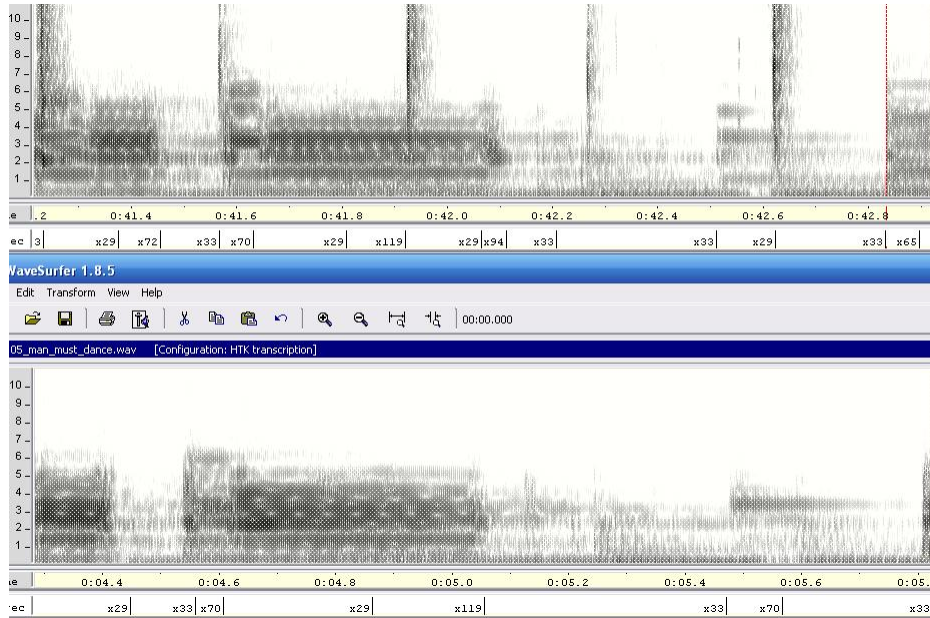


Figure 4.2: Example of ASM tokenization for two similar melodies as a solo (lower waveform) and with polyphonic ornamentation (upper waveform).

4.2 Musical Chord Recognition

In recent years, automatic chord recognition has received increased attention from researchers because the harmonic analysis provides an important mid-level representation of music. Unlike transcription tasks, chord detection algorithms do not need to know the number of sources *a priori*. Further, most systems do not require error-prone source separation techniques. Most recent approaches to identifying chords from the acoustic signal use chroma-based features called pitch class profiles (PCP) [38] (see 4.2.1) as inputs into an HMM-based system. For example, [56] used an ergodic HMM to provide an initial chord progression, which was updated using an N -best list. Because transcribing chord progressions manually is an expensive task, research has focused on achieving robust models with limited training data. Sheh and Ellis [114] assume PCP vectors from the same mode (e.g., *Major*) can be considered as rotated versions of one another. The PCP vectors are rotated to the same root note and used to estimate a single mode-specific density. To develop densities for each

pitch class, the mode-level densities are rotated back to each pitch class. Lee and Slaney [64] create synthesized audio to increase the amount of training data. Bello *et al.* [12] incorporate musical knowledge to update the HMM parameters because of limited training data. Recently, discriminative approaches, such as MCE training [103] and SVMstruct [133], have been utilized because a chord is most often misclassified by a particular competing chord; e.g., a C-Major chord is most often confused with either a G-Major or C-minor chord.

In Chapter 3, it is stated that ASMs are comparable to phonemes. This section demonstrates this interpretation by using ASMs to produce chord transcriptions. Specifically, a set of ASMs is trained on a private dataset and then used to produce an unsupervised transcription on PCP vectors. Next, the resulting ASMs transcribe a small amount of labeled training data in order to build a dictionary of “chord pronunciations.” Currently, the only freely available chord database is The Beatles catalog with chord labels provided by Harte [45]; therefore, both the training data and test data are from the same artist. MAP adaption [41] is utilized in a structured format [115] to shift the ASMs to match the training and test conditions.

4.2.1 Chord Recognition Features

PCP vectors (or chromagrams) [10] are a mapping of the energy spectrum to pitch class (or chroma) energy and are the most common feature used for automatic chord recognition; however, it has been noted that PCPs are susceptible to transients from onsets and percussive instrumentation [91]. The harmonic/percussive source separation (HPSS) algorithm [85], which has been successfully applied to chord recognition [128][103], is used to remove the percussive transient effects by minimizing

$$J(\mathbf{H}, \mathbf{P}) = \frac{1}{2\sigma_H^2} \sum_{k,n} (H_{k,n-1} - H_{k,n})^2 + \frac{1}{2\sigma_P^2} \sum_{k,n} (P_{k-1,n} - P_{k,n})^2, \quad (4.1)$$

where $H_{k,n}$ and $P_{k,n}$ are the values of the power spectrum at frequency index k at time index n for the harmonic spectrum, \mathbf{H} , and the percussive spectrum, \mathbf{P} , respectively.

The parameters σ_P and σ_H are set using a cross-validation set. In addition, constraints are placed such that for each index k and n , the sum of the harmonic and percussive spectrum sum to the original spectral value at frequency k and time index n . Finally, both the harmonic and percussive spectrum must be non-negative. Further details of the HPSS algorithm can be found in [85].

Only the harmonic spectrum is retained for calculation of the PCP vectors. The harmonic portion of the original audio spectrum is down-sampled to 11025 Hz and broken into frames of 2048 samples with a 50% overlap between successive frames. The constant-Q spectrum [22] computes the audio spectrum, S , of the audio signal, $s(t)$, by

$$S(k) = \sum_{t=0}^{T(k)-1} w(t, k) s(t) e^{-j2\pi f_k t}, \quad (4.2)$$

where $w(t, k)$ and $T(k)$ are functions of the frequency bin index, k . The frequency bin locations are such that the center frequencies are

$$f_k = 2^{k/\beta} f_{ref}, \quad (4.3)$$

where f_k is the center frequency for the k^{th} bin, β is the number of bins per octave, and f_{ref} is the reference frequency. To have each bin match a musical note frequency on the equal-temperament scale, β is set to 12 and f_{ref} is set to 27.5 Hz, which is the frequency of note $A0$. The traditional PCP feature vector is given by

$$c(i) = \sum_{\phi=0}^{\Phi} |S(i + \phi\beta)|, \quad (4.4)$$

where $i = \{1, 2, \dots, \beta\}$ is the pitch class bin number and Φ is the number of octaves considered. Because harmonics of different pitch classes overlap, PCP vectors are highly correlated [12]; therefore, a DFT is applied on the PCP vectors to reduce the cross-correlation, as demonstrated in [103].

Table 4.3: Example ASM sequences for chords.

Chord	ASM Sequence	Probability
D-Major	x29 x24	0.27
	x29 x49 x24	0.13
	x24	0.60
D#-Major	x57	1.0
E-Major	x32	0.18
	x53	0.63
	x18	0.19

4.2.2 ASMs to Chords

The PCP representation of a private dataset, which has no overlap with the training and test sets, is used to train the ASMs. Each ASM is represented with a single-emitting state HMM, where each emitting state contains an eight-mixture GMM density with diagonal covariances. The choice of using a single-state HMM is to match the baseline approach. In addition, it is noted that the data frames for PCP vectors (0.185s with 0.0925s overlap) are much larger than for MFCCs (0.025s with 0.0125s overlap).

Each song in the training and test set contains a ground-truth chord transcription with both the chord and timing information. The training songs are segmented into individual chords and each individual chord is decoded with the ASM set. Note that a chord may be decoded with a sequence of ASM tokens, as shown in Table 4.3; therefore, each chord may be viewed as similar to a word in speech and each ASM token may be thought of as a phoneme. A set of ASM “pronunciations” is produced for each chord, along with a probability for each chord pronunciation, to build a chord dictionary. To prevent spurious pronunciations, only ASM sequences that have a probability above a certain threshold (10%) are considered for further processing.

Because the training and testing sets are from the same artists, structured-MAP (SMAP) adaptation is performed to update the HMM models to compare with the baseline. The SMAP algorithm was proposed in [115] for speaker adaptation because

it is well-known that the performance of speech recognition algorithms degrade when there is a mismatch between the training and testing conditions. A common source of this mismatch is different speakers. Systems that have the same speaker for both testing and training conditions are called “speaker-dependent” systems. Gathering enough data from a single speaker to train all the necessary model parameters for a maximum likelihood estimate is often too costly. Therefore, a small number of recordings from several speakers is often used to train a single “speaker-independent” model set. However, this creates a mismatch between the training and testing conditions.

One technique to account for this mismatch is to assume that the model parameters are a random variable with an assumed joint probability density function. Using the data from a single speaker, a Bayesian estimate is found for the posterior distribution by assuming the speaker-independent model defines a prior probability density function for the speaker-dependent model. This training procedure, known as MAP adaptation [41], has been shown to be effective and is asymptotically equivalent to MLE. A weakness of MAP adaptation is that models not observed in the adaptation data are not updated; therefore, SMAP defines a hierarchy on the model parameters to guide the adaptation process.

SMAP is a two stage process of defining a hierarchical structure of Gaussians and then performing MAP adaptation by assuming that the prior density for a given node is the parent of the given node. To build the hierarchy, first all the Gaussians from all the mixture densities from all the HMMs are grouped at the root node. Next, the ASM Gaussians in the current node are approximated by a node probability density function, which is also chosen to be a Gaussian density. Next, the symmetric Kullback-Leibler divergence defines a distance metric from a given ASM Gaussian and a root node probability density function. Using this distance measure, the ASM Gaussians are then clustered into a set of children nodes. The children nodes are also modeled as a Gaussian node density. The parameters of the child node density

are an interpolated average between the estimated density using the ASM Gaussians assigned to the child node and the parameters of the parent node. Each child then serves as the parent node to a subsequent tree level. The second stage of SMAP uses the tree structure to update the model parameters by assuming that a parent node is the conjugate prior to each of its children nodes.

The creation of the ASM pronunciation dictionary and SMAP adaptation is iterated three times. That is, an initial dictionary is created and ASM sequences are dropped if their probability of occurring for a particular chord is less than 10%. SMAP is then performed. The training set is then decoded using the updated ASMs and a new dictionary is created. This process is repeated twice more. A bigram language model is estimated from the training data.

4.2.3 Baseline Chord Recognition System

The baseline chord recognition system is the system that performed best at the 2008 MIREX Audio Chord Detection contest. The same PCP vectors used in the acoustic segment modeling approach are used for the baseline system. Each chord is modeled with a single-emitting state HMM with a Gaussian observation density. MLE is performed using the training data. The most likely chord sequence to produce a test song is found using Viterbi decoding. The same bigram language model as the acoustic segment modeling system is used.

4.2.4 Data and Evaluation

The training and test data is The Beatles catalog, which was transcribed by Harte [45]. A total of ten evaluations is performed, where for each evaluation, five albums are randomly selected for training, five are randomly selected for testing, and there is no album overlap between the training and testing data. The private data used to train the ASMs consists of 600 songs from the author's collection and does not contain any songs by The Beatles or a Beatles member. Further, songs by The Beatles or a

Beatles member, but performed by a different artist or group (i.e., cover songs) were also removed from the private dataset.

The evaluation metric used is frame error rate (FER) and the percent accuracy given by the HTK toolkit [142]. The FER is simply

$$\text{FER} = \frac{\text{number of frames correct}}{\text{number of frames}}. \quad (4.5)$$

Note that FER is dependent on the number of frames and the frame length because chord boundaries do not necessarily align with frame size. Therefore, a standard window and hop size must be given if a system does not find note boundary locations prior to segmenting the audio file. A related measure to FER is percent overlap and is also very dependent on chord boundaries. However, accurately locating chord boundaries is a difficult and somewhat subjective task. Therefore, even very small changes in an annotated frame boundary can greatly affect FER or percent overlap [103]. The ASR community has generally avoided the similar problem of word boundary locations by focusing on measures that penalize insertion, substitution, and deletion errors, but ignore boundary locations. One such measure is the percent accuracy, which is found by finding the optimal string alignment between the predicted and reference transcriptions, and is given by (3.2).

4.2.5 Results

To demonstrate how SMAP improves performance, the chord recognition accuracy for isolated chords is presented for the first training and testing fold. This was accomplished by isolating each chord based on the chord labels provided. As can be seen in Table 4.4, the isolated recognition rates improve after adapting the “artist-independent” models to match the training and test set. When the amount of speaker-dependent data used to train the maximum likelihood models is the same amount of data used to perform MAP adaptation from a speaker-independent density, it is well-known that MAP adaptation performs better in speech recognition tasks [41].

Table 4.4: Isolated chord accuracies for the first training and testing set.

Iteration	Train	Test
1	0.749	0.729
2	0.781	0.759
3	0.811	0.768

This is especially when the amount of adaptation data is small. However, it was not possible to test whether this occurs for chord recognition because it is necessary to have enough data to build an ASM-chord dictionary. Ideally, the ASM-chord dictionary would be estimated from several different artists and then the models would be adapted.

Regardless, the ASM-chord dictionary performs surprisingly well in terms of FER, as shown in Table 4.5. It should be emphasized that the baseline was also the best approach at the 2008 MIREX Audio Chord Detection contest². In fact, from Table 4.6 it is seen that deletion errors are a particular problem for the acoustic segment modeling approach. A better ASM-chord dictionary may be able to improve results in the future. Further, the goal of this section is not to improve on the audio chord detection task, but to demonstrate the ability of the unsupervised acoustic segment modeling procedure to capture temporal information.

4.3 Summary

This chapter demonstrates how the acoustic segment modeling procedure presented in Chapter 3 is able to detect temporal structures in music. First, it is demonstrated how ASMs can detect temporal semantic tags better than a state-of-the-art spectral-based approach [125]. The system trains a set of ASMs in an unsupervised process and models each ASM with an HMM. Next, LSA vectorizes a given ASM transcription by forming a vector of ASM unigram and bigram counts. Sparsity is

²http://www.music-ir.org/mirex/wiki/2008:Audio_Chord_Detection_Results

Table 4.5: FER for baseline and ASM approach.

Set	ASM	Baseline
1	29.38	25.53
2	30.54	27.83
3	27.82	23.58
4	25.77	20.50
5	27.81	24.81
6	25.26	23.16
7	27.46	23.87
8	34.73	31.99
9	23.97	20.49
10	28.27	25.80

Table 4.6: HTK results for the chord detection task.

	ASM	Baseline
Acc %	57.40	60.03
Deletions	14922	10050
Substitutions	6625	6784
Insertions	2042	5298

then reduced by SVD, and the dimensionally reduced vectors serve as training data into a bank of binary SVMs, where each SVM is trained to detect to presence of a particular attribute. A notable and significant improvement is seen over the baseline GMM approach, especially for tags describing temporal aspects in music. Further improvement may be possible by extending the count vectors to include higher-order statistics [104], e.g., trigrams, quadgrams, etc.

The ability of ASMs to detect temporal structure in music is further seen on a chord detection task. A set of ASMs are trained in an unsupervised process. The most likely ASM sequence to produce each chord is identified by using a small amount of labeled training data. Each ASM sequence can be equated to a pronunciation of the chord. Due to the fact that the training and testing data are from the same artist, MAP adaptation in a structured format is performed. Results indicate that the acoustic segment modeling procedure performs almost as well as the baseline

maximum likelihood approach.

The procedure of mapping PCP-based ASMs to musical chords is only in an initial stage of research. In fact, the acoustic segment modeling approach is likely to perform best when several artists are used to build the chord dictionaries. However, current labeled training and testing data [45] is composed of a single artist, i.e., The Beatles. From the viewpoint of ASR, the acoustic segment modeling procedure can be seen as a way to perform unsupervised speaker-independent speech recognition because the data comes from several artists. However, the task of musical chord detection with existing databases is similar to a speaker-dependent speech recognition system. Further, the goal of this chapter is to demonstrate that ASMs are able to capture a large portion of the harmonic structure and not to perform the best possible on this task. Potentially, the approach in this section could be used to increase robustness of chord recognition tasks, but this is currently left for further research. In Chapter 7, the chord-based ASMs are combined with the MFCC-based ASMs to yield improved performance in content-based analysis.

CHAPTER V

MUSIC SIMILARITY FOR CONTENT-BASED ALGORITHMS

In recent years, many authors have questioned the utility of automatic genre recognition and music similarity [4][5][14][76][87]. Indeed, music genre categories largely arose based on marketing reasons and at a time when The Long Tail had never been contemplated [27]. Since this taxonomy did not arise based on theoretical concerns, no standard was ever achieved; therefore, genres are constantly in flux, with new genres arising, old genres dying or splitting, and boundaries between genres being generally vague [4].

Such poor labeling schemes impact content-based approaches to performing genre recognition and music similarity. Despite recent progress in genre recognition [90], current databases are limited in the number of genres; e.g., typically, ten genres are considered. Further, the genres often form a very coarse taxonomy, e.g., rock, country, classical, etc. In addition, as these systems are further optimized on a given dataset, they exhibit poor generalization ability to new datasets, even if the new dataset resembles the old dataset in character [88]. One reason is that the different datasets use different categorizations, so that even the same song may be labeled differently across datasets [4]. While some authors have tried to investigate other music similarity classification schemes [87][14], it was noted by McKay and Fujinaga [76] that music similarity in general suffers from the same subjective issues as genre.

One potential cause suggested by researchers is that high-level concepts, such as genre, are inherently subjective and not entirely based on the acoustic content. The implication is that non-acoustic cues bias a person's categorization of genre. For

example, simply noting an artist’s clothing and hairstyle elicits an expectation of the type of music to be performed [50]. This ultimately results in the futility of developing systems for content-based retrieval because the metrics are unknown in their character and upper limits. For example, if an accuracy measure of 80% is cited as an upperbound based on results from a user study, does this mean that the system is 80% effective against a general population, a population with certain characteristics, a single person, or does it have an accuracy of 100% for four out of five people and gets nothing right for the fifth person? Without a well-defined metric, comparing results becomes very difficult.

To address these concerns, this chapter attempts to answer whether it is possible to define traditional music categories based on musical attributes. That is, does a group of musicologists associate various attributes with their own genre assignments? As a comparison, an artificial taxonomy is constructed by clustering songs based on musical attributes. The two taxonomies are compared in a proven discriminative-training classifier, which was originally designed for natural call routing [62]. The choice of this particular algorithm is based on its effectiveness in a natural language classification task that is potentially subjective and where appropriate features are not obvious. Also, optimization is an easier task than many constraint-based algorithms because the objective function is differentiable and solvable through the use of gradient probabilistic descent (GPD) [54].

5.1 Song Description Vector

This chapter differs from the rest of this dissertation in that no acoustic analysis is performed. Instead, text documents represent the initial data, which come in the form of a musical attribute list. Specifically, the dataset is the set of Pandora song descriptions for the songs used in Section 4.1. Since Pandora contains only about 500 attributes, a vocabulary of 375 musically relevant words was manually constructed

from the dataset. The words were chosen in a very conservative fashion to ensure an important word was not omitted. Therefore, word selection was manual, since a few commonly omitted stop words (e.g., *off*) can have a musically relevant meaning (e.g., *off beat*). Each song description vector is created using LSA (see Section 3.4.2.1), so that each document is converted to a vector of weighted word counts and then projected into a lower-dimensional “concept space” [11]. In this “concept space,” similar documents retain proximity even when they do contain any of the same words.

Further, many of the attributes are genre descriptions, which are used as ground-truth labels for the genre classifier. This ensures that the genre labels arise from the same source as the input data. Since Pandora’s experts go through a long training regimen to maintain consistency [51], it is assumed that all song descriptions on Pandora are from a highly consistent and reliable source. Obviously, a better dataset would arise from a single source (i.e., a single musicologist), but the information about which musicologist edited a song description is not provided. In addition, it takes 20-30 minutes to annotate a single song¹, which would make any dataset using a single editor too small to reliably extract results. While 25 different genres were found, only genres with 40 or more songs were retained, which resulted in the following genres: country, pop-rock, rap, r&b, and rock. A test set of 5825 songs from artists not contained in the USPop dataset was also constructed. Weighted word count test vectors are projected into the “concept space” using (3.10).

¹<http://blog.pandora.com/faq/contents/506.html>

5.2 Attribute-based Taxonomy

To evaluate the genre-based taxonomy, an upper-bound is created by developing a new taxonomy based on how the attribute documents cluster in the lower-dimension “concept space.” Similarity is determined by the cosine between two song description vectors:

$$d_{cos}(\mathbf{x}_i, \mathbf{x}_j) = \cos(\mathbf{v}_i \Sigma_{\rho_0}, \mathbf{v}_j \Sigma_{\rho_0}) = \frac{\mathbf{v}_i \Sigma_{\rho_0}^2 \mathbf{v}_j^T}{\|\mathbf{v}_i \Sigma_{\rho_0}\| \|\mathbf{v}_j \Sigma_{\rho_0}\|}, \quad (5.1)$$

where \mathbf{v}_i and \mathbf{v}_j are two dimensionally reduced song vectors from V_{ρ_0} and $\|\cdot\|$ represents the L_2 norm.

Next, a bottom-up clustering procedure groups similar song descriptions such that each song description vector, \mathbf{x}_i , in a cluster has a similarity of $d_{cos}(\mathbf{x}_i, \hat{\mathbf{x}}_k) < \epsilon$, where ϵ is a similarity threshold and $\hat{\mathbf{x}}$ is the cluster mean. These clusters are based on the musical descriptions given by Pandora and are free from non-acoustic features. Clusters that contained more than 10 songs are retained. As an illustrative example, one cluster contained the following songs:

1. “Tiger” by Abba
2. “The Ballad of El Goodo” by Big Star
3. “Flowers” by New Radicals
4. “Simple Kind of Life” by No Doubt
5. “How’s it Going to Be” by Third Eye Blind
6. “She Takes Her Clothes Off” by Stereophonics

These songs all have similar attributes, such as *basic rock song structure, mixed acoustic and electric instrumentation, a vocal harmony, and major key tonality.*

5.3 Discriminative-Training Song Classification

This section describes the classifier used for the experiments in Section 5.4. Given a set of $|\mathcal{C}|$ categories, $\mathcal{C} = (\mathbb{C}_1, \mathbb{C}_2, \dots, \mathbb{C}_{|\mathcal{C}|})$, the goal is to classify a given test document into the correct class using the N training documents. First the misclassification function $d_k(\mathbf{x})$, assigns each ρ_0 -dimensional song description vector according to

$$\hat{k} = \arg \min_{k \in \mathcal{C}} d_k(\mathbf{x}) = \arg \min_{k \in \mathcal{C}} [-g_k(\mathbf{x}) + G_k(\mathbf{x})]. \quad (5.2)$$

The function $g_k(\cdot)$ is called the discriminate function and is the dot product between \mathbf{x} and the k^{th} class vector, *mathbf{w}_k*:

$$g_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x}. \quad (5.3)$$

The function $G_k(\cdot)$ is called the anti-discriminate function and is

$$G_k(\mathbf{x}) = \left[\frac{1}{|\mathcal{C}| - 1} \sum_{i \neq k, 1 \leq i \leq |\mathcal{C}|} g_i(\mathbf{x})^\eta \right]^{1/\eta}, \quad (5.4)$$

where η is a positive number and determines the importance of the competing classes. In particular, as $\eta \rightarrow \infty$, G_j is dominated by the most competitive class.

Note that the misclassification function given in (5.2) is negative if \mathbf{x} is predicted to be in the k^{th} class and positive if \mathbf{x} is predicted to be in another class. The confidence of the decision is proportional to the magnitude of $d_k(\mathbf{x})$; therefore, a loss function can be defined using the misclassification function. Ideally, such a loss function should approximate a 0-1 loss function. A common choice in discriminative-training techniques is to use a sigmoid function (see Figure 5.1):

$$l_k(\mathbf{x}) = \frac{1}{1 + \exp(-\alpha d_k(\mathbf{x}) + \gamma)}, \quad (5.5)$$

where α and γ are parameters that control the slope and shift of the sigmoid function, respectively. Note that this loss function is continuous and differentiable; therefore,

the empirical error becomes continuous and differentiable when the 0-1 loss is replaced by (5.5):

$$L_{emp}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^{|\mathcal{C}|} l_k(\mathbf{x}_i) 1(\mathbf{x}_i \in \mathcal{C}_k), \quad (5.6)$$

where $1(\cdot)$ is the indicator function. Note that if $l_k(\mathbf{x})$ is a 0-1 loss then (5.6) is the empirical error. Because the approximation of the empirical error is continuous, efficient solutions can be found using GPD.

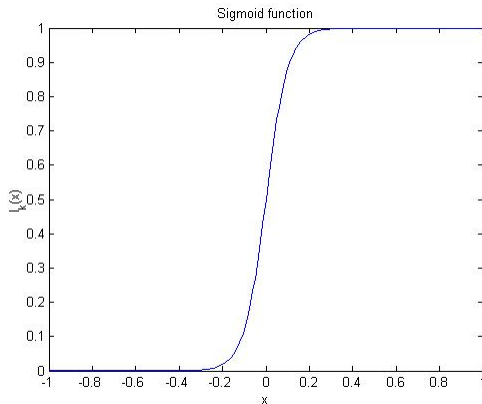


Figure 5.1: Sigmoid-based loss function.

5.4 *Experimental Results*

An example mean vector for the artificial taxonomy is shown in Figure 5.2 and demonstrates the effect of the discriminative-training algorithm on the term weights. From a qualitative standpoint, the terms with the biggest positive and negative weights indicate that this class contains many songs with acoustic guitar riffs and do not feature breathy vocals, antiphony, or minor tonality. Therefore, the discriminative-training algorithm is able to emphasize features that best discriminate the given cluster or genre, while also highlighting features that negatively correlate with the cluster or genre.

The training loss defined in (5.6) is shown in Figure 5.3 using the genre labels and the artificial taxonomy. As can be seen, the genre labels are much more difficult to

train than the baseline that serves as an upper bound. In fact, the genre labels do not converge until after 16,000 iterations to a value of about 4%, which is still much higher than the attribute-based taxonomy. By comparison, the artificial taxonomy is quite easy to learn, which indicates that songs can be clustered reliably based on acoustic attributes.

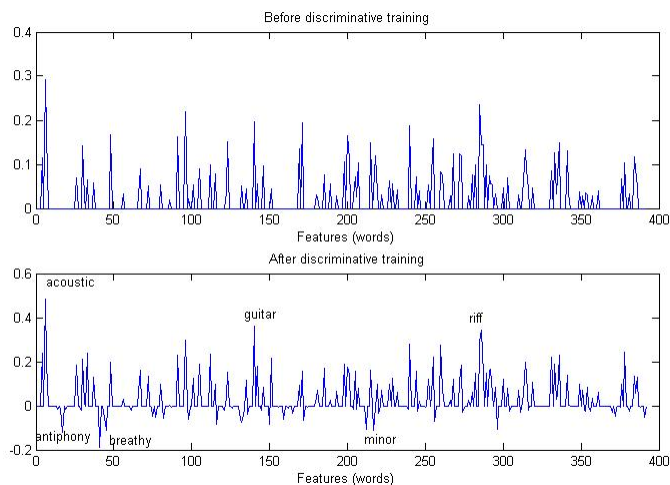


Figure 5.2: Attribute weights before and after the discriminative-training procedure for a chosen cluster.

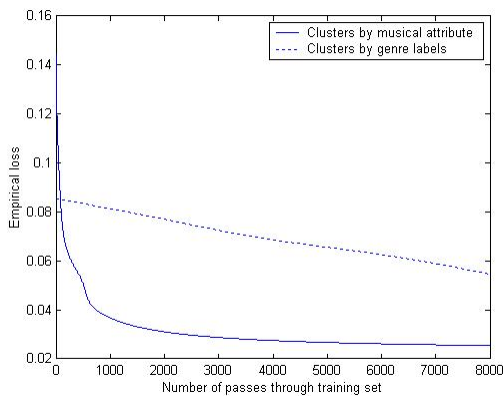


Figure 5.3: Empirical loss for the discriminative-training song description clusters.

Table 5.1: Confusion matrix for classifying genres by Pandora attributes.

Genre	country	pop-rock	r&b	rap	rock
country	49.65	2.8	2.45	0	45.1
pop-rock	1.81	15.03	8.81	0	74.35
r&b	0	14.3	38.12	0	47.54
rap	3.31	0.74	2.21	87.5	6.25
rock	1.34	4.72	1.68	0.09	92.16

The inability for genre labels to correlate with the acoustic attributes is further confirmed on the test set, where the genre classification accuracy is 80.26%. The confusion matrix is shown in Table 5.1. These results indicate that even musicologists are unable to correlate acoustic descriptors with genre taxonomies. If the gap between mid-level cognitive features cannot describe the high-level attribute of genre association, there is little hope that low-level acoustic features will perform better. These results should not be interpreted as saying that genre recognition is an impossible task. Instead, this indicates that non-acoustic and even personal information may be necessary in order to accomplish the genre recognition task.

5.5 *Summary*

Previous content-based MIR algorithms have typically assumed a taxonomy that was universal; however, results have forced researchers to investigate this assumption. Ultimately, content-based algorithms are limited by the assumptions placed on the data, the model, and the labels. First, content-based algorithms rely on extracting features that are able to highlight important similarities and differences in the data while removing noise. Second, modeling assumptions heavily impact classification performance. For instance, it was shown in Chapter 4 that the “bag-of-frames” assumption limited the ability to model temporal information in the music signal. Finally, this chapter demonstrated that supervised approaches rely on correct labels

to adequately model classification boundaries. Labeling errors ultimately place a performance ceiling on any supervised approach. Further, error metrics become difficult to understand qualitatively when it is unclear as to how given classes relate to one another. Most importantly, content-based algorithms rely on the assumption that all information necessary for discriminating classification categories is contained in the content of the signal.

To investigate these questions, this chapter investigated whether any consistent taxonomy could be derived from an acoustic signal. Specifically, it was investigated whether acoustic attributes consistently align with a musicologist’s viewpoint of genre. To maintain consistency, the Pandora Music Genome Project was utilized because all annotators are specially trained to maintain labeling consistency. The results presented in this chapter demonstrate that acoustic attributes do not define genre consistently when both the attributes and genre labels arise from the same source. The artificial taxonomy is only used to illustrate that a system can find reliable performance if only acoustic attributes are detected in a binary decision. Potentially, this artificial taxonomy could be used to guide a content-based algorithm; however, such a system requires the user to learn the new taxonomy.

An important implication is that concepts of genre rely on information exterior to the acoustic information. As an example, social information can often override acoustic similarity. For example, Pandora lists eight attributes for teen-pop star Justin Bieber’s “Down to Earth.” A comparison with the artist’s collection finds that The White Stripes’s “This Protector,” Wilco’s “I’m the Man Who Loves You,” and The Polyphonic Spree’s “Section 12 (Hold Me Now)” share six, six, and all eight attributes, respectively. Very few people would link any of these bands to Justin Bieber, despite their similar acoustic attributes. This means that the genre recognition task is an ill-defined task for content-based retrieval, even when the task is designed to provide individual genre labels. This result is not to be interpreted

as saying that genre recognition and music similarity are ill-defined tasks in general; rather, more information than is provided in the acoustic signal is necessary to address these tasks. The next chapter addresses this issue of subjectiveness by providing personalized content-based retrieval.

CHAPTER VI

CONTENT-BASED PREFERENCE RANKING

The results in Chapter 5 demonstrate that attempts to link low-level acoustic features to high-level cognitive concepts such as genre are unlikely to succeed. This is because more measurable cognitive concepts that are easier to model, such as *repetitive melodic phrasing*, *use of syncopation*, etc., do not describe genre well. Two noted reasons are the subjectiveness of many high-level concepts and that musical similarity is often shaped by factors exterior to the acoustic signal. One implication is that categorization is likely personal in nature and difficult to generalize across a population.

One such personalized categorization is ordinal scales, such as assigning a song “4 out of 5 stars.” In fact, most online radio stations use some degree of preference rating, e.g., buttons for “thumbs up” or “thumbs down.” This chapter investigates content-based personal ratings prediction and presents a novel ordinal regression algorithm motivated by the discriminative-training technique known as MCE training [54] and is related to maximal figure-of-merit (MFoM) classifier [40]. Further, this chapter investigates whether incorporating temporal information results in an improvement over spectral-based approaches. In total, three spectral-based approaches will be compared with the acoustic segment modeling procedure. All three are based on GMM modeling, but differ in how densities are estimated and ultimately converted into a vector.

6.1 Discriminative-Training Ordinal Regression

As mentioned in Section 2.4, the objective function of ordinal regression is to minimize the average ranking loss (ARL) in (2.6). However, (2.6) is a discrete function and difficult to optimize. The motivation of the approach presented is to view the ARL as the misclassification rates summed against all the decision boundaries:

$$L_{est}(\mathcal{X}; \mathbf{w}, \mathbf{b}) = \sum_{r=1}^{R-1} L_{r+} + L_{r-}, \quad (6.1)$$

$$L_{r+} = \frac{1}{|L_{r+}|} \sum_{i=1}^{|L_{r+}|} l_{r+}(\mathbf{x}_i; \mathbf{w}, b_r) \mathbf{1}(\mathbf{x}_i \in \mathbb{C}_{r+}), \quad (6.2)$$

and

$$L_{r-} = \frac{1}{|L_{r-}|} \sum_{i=1}^{|L_{r-}|} l_{r-}(\mathbf{x}_i; \mathbf{w}, b_r) \mathbf{1}(\mathbf{x}_i \in \mathbb{C}_{r-}), \quad (6.3)$$

where \mathbb{C}_{r+} is the set of $|L_{r+}|$ positive training samples for decision boundary r , \mathbb{C}_{r-} is the set of $|L_{r-}|$ negative training samples for decision boundary r , and $\mathbf{1}(\cdot)$ is the indicator function. Note that the class membership can contain multiple ranks. For example, if there are five possible ranks ($r = \{1, 2, \dots, 5\}$) and the boundary considered is between $r = 2$ and $r = 3$, \mathbb{C}_{r+} contains any training samples such that $r \geq 3$ and \mathbb{C}_{r-} contains training samples such that $r < 3$. The loss function, $l_r(\mathbf{x}; \mathbf{w}, b_r)$, is traditionally the 0-1 loss, which is not differentiable.

MCE builds a continuous approximation for the discrete objective measure of misclassification rate by modeling the 0-1 loss with a smooth and differentiable function. Ideally the continuous approximation should have a value close to zero for a correct classification and one for an incorrect classification. A common choice for the loss function is the sigmoid function,

$$l_r(\mathbf{x}; \mathbf{w}, b_r) = \frac{1}{1 + e^{-[\alpha d_r(\mathbf{x}; \mathbf{w}, b_r) + \gamma]}}, \quad (6.4)$$

where α controls the slope and γ controls the offset. The function $d_r(\mathbf{x}; \mathbf{w}, b_r)$ is designed such that (6.4) approaches zero as the a sample is classified correctly with

greater confidence and approaches one as the sample is classified incorrectly, but with more confidence in the incorrect decision:

$$d_r(\mathbf{x}; \mathbf{w}, b_r) = \begin{cases} -\mathbf{w}^T \mathbf{x} + b_r & \mathbf{x} \in \mathbb{C}_{r+} \\ \mathbf{w}^T \mathbf{x} - b_r & \mathbf{x} \in \mathbb{C}_{r-} \end{cases}. \quad (6.5)$$

Note that as $\alpha \rightarrow \infty$, (6.4) approximates a 0-1 loss, which makes (6.1) a continuous and differentiable approximation for the discrete objective given in (2.6). This allows for the optimization of a smoothed version of the ARL without constraints by using GPD. The process is an iterative procedure, where at each iteration n , the update rules for \mathbf{w} and $\mathbf{b} = [b_1, b_2, \dots, b_{R-1}]$ are

$$\mathbf{w}^{n+1} = \mathbf{w}^n - \kappa_n \left. \frac{\partial L(\mathbf{x}; \mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_n, \mathbf{b}=\mathbf{b}_n} \quad (6.6)$$

and

$$b_r^{n+1} = b_r^n - \kappa_n \left. \frac{\partial L(\mathbf{x}; \mathbf{w}, \mathbf{b})}{\partial b_r} \right|_{\mathbf{w}=\mathbf{w}_n, \mathbf{b}=\mathbf{b}_n}, \quad (6.7)$$

where κ_n is the learning rate at iteration n . The partial derivatives are given by

$$\frac{\partial L(\mathcal{X}; \mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = \sum_{r=1}^{R-1} [L_{\mathbf{w}+} + L_{\mathbf{w}-}], \quad (6.8)$$

$$L_{\mathbf{w}+} = \frac{1}{|L_{r+}|} \sum_{i=1}^{|L_{r+}|} \alpha l_{r+} (1 - l_{r+}) (-\mathbf{x}_i) 1(\mathbf{x}_i \in \mathbb{C}_{r+}), \quad (6.9)$$

$$L_{\mathbf{w}-} = \frac{1}{|L_{r-}|} \sum_{i=1}^{|L_{r-}|} \alpha l_{r-} (1 - l_{r-}) (\mathbf{x}_i) 1(\mathbf{x}_i \in \mathbb{C}_{r-}), \quad (6.10)$$

$$\frac{\partial L(\mathcal{X}; \mathbf{w}, \mathbf{b})}{\partial b_r} = \sum_{r=1}^{R-1} [L_{b_{r+}} + L_{b_{r-}}], \quad (6.11)$$

$$L_{b_{r+}} = \frac{1}{|L_{r+}|} \sum_{i=1}^{|L_{r+}|} \alpha l_{r+} (1 - l_{r+}) (-1) 1(\mathbf{x}_i \in \mathbb{C}_{r+}), \quad (6.12)$$

and

$$L_{b_{r-}} = \frac{1}{|L_{r-}|} \sum_{i=1}^{|L_{r-}|} \alpha l_{r-} (1 - l_{r-}) (1) 1(\mathbf{x}_i \in \mathbb{C}_{r-}). \quad (6.13)$$

Note that the decision boundaries are coupled in the update equations through \mathbf{w} , which is the direction that best discriminates the ranks. Further, each offset, b_r , is updated to minimize the misclassification rate among all the training samples and not just the neighboring ranks.

At each iteration, the logistic offset, γ , is updated to be the average of (6.5) taken across all the ranks and training samples. The slope of the logistic function, α , is found through cross-validation (see Section 6.5). While the learning rate, κ , can vary over time, it was held to a constant value of 0.01.

6.2 Baseline Ranking Algorithm: PRank [29]

The baseline ranking algorithm for this chapter is PRank [29], which is a ranking algorithm based on the perceptron algorithm. Like the discriminative-training ranking algorithm, the goal is to learn a function of the form in (2.5) such that the ARL in (2.6) is minimized. The major difference between PRank and the discriminative-training ranking algorithm given in (2.6) is that PRank is a *conservative* [29] learning algorithm because it only learns from training mistakes.

PRank starts by noting that a training sample, \mathbf{x}_i , is correct if

$$\mathbf{w}^T \mathbf{x}_i \begin{cases} > b_r & 1 \leq r \leq y_i - 1 \\ < b_r & y_i \leq r \leq R \end{cases}, \quad (6.14)$$

where y_i is the true rank of and $b_R = \infty$. Like the perceptron algorithm, PRank first projects the data to the reals; however, instead of comparing to a single threshold, multiple thresholds are evaluated and the final value is the minimum rank where the projected value is below the threshold. If a mistake is incurred then there is at least one rank, r , where an error occurred. The perceptron learning rule is used to move the projected value, $\mathbf{w}^T \mathbf{x}_i$, and b_r towards one another. Specifically, if training sample \mathbf{x}_i is truly assigned a rank $y_i \neq r$, but is misclassified by PRank as rank r ,

the update rule is

$$b_r^{new} = \begin{cases} b_r^{old} - 1 & \mathbf{w}^T \mathbf{x}_i < b_r^{old} \\ b_r^{old} + 1 & \mathbf{w}^T \mathbf{x}_i > b_r^{old} \end{cases} \quad (6.15)$$

and

$$\mathbf{w} = \mathbf{w} + \left(\sum_{r \in \mathcal{I}} \tau_r \right) \mathbf{x}_i, \quad (6.16)$$

where \mathcal{I} is the set of ranks that are misclassified and τ_r is -1 if $y_i \leq r$ and $+1$ if $y_i > r$.

6.3 Data Modeling

The previous section detailed the two ranking algorithms that are compared in this chapter. Since both algorithms are vector-based, a vector representation of the audio is needed. Four representations are given in this section. The first is the acoustic segment modeling procedure, which is a tokenization approach. The last three are spectral-based approaches that model an acoustic object with a GMM or a Gaussian density under the “bag-of-frames” assumption.

6.3.1 Acoustic Segment Modeling Specifications

The acoustic segment modeling procedure in Chapter 3 is followed. Each ASM is modeled with an HMM with three-emitting states, and each state uses an eight-mixture GMM with diagonal covariance matrices. The final ASM transcripts are converted to vectors of weighted word counts using LSA (see Section 3.4.2.1). Since a total of 128 ASMs are used, the resulting vectors are of length 16,512 because unigrams and bigrams are considered. To reduce the dimensionality and sparsity of the data, the final step of LSA uses SVD to produce the dimensionally reduced matrix, V_{ρ_0} , which serves as the training data for the PRank algorithm and the discriminative-training algorithm, denoted as ASM/PRank and ASM/DTRank, respectively.

6.3.2 Spectral-based approach 1: Single Gaussian

The first spectral-based approach models each song or artist with a single Gaussian, where the covariance matrix is a full matrix. A Gaussian density is converted to a vector by concatenating the MFCC mean and unwrapped covariance values. Each song or artist vector has a dimension of $D + D^2$, where D is the dimensionality of the data. The vectors produced with this procedure serve as the inputs into the PRank algorithm and the discriminative-training algorithm and are denoted as MeanVar/PRank and MeanVar/DTRank, respectively.

6.3.3 Spectral-based approach 2: Bag-of-Timbres

The second spectral-based approach assumes each song or artist can be modeled as a “bag-of-timbres” [6][73][141]. First, a GMM models the entire acoustic space by using all the songs in the training database. Such a GMM is known as a universal background model (UBM) in the ASR community [105]. However, because the dataset is quite large in terms of duration, ten seconds are randomly selected for each song, which provided good results on the cross-validation set (see Section 6.5).

The parameters of the UBM are estimated by the maximum likelihood criterion:

$$\hat{\theta} = \arg \max_{\theta} \log \prod_{i=1}^N f(\mathbf{x}_i | \theta), \quad (6.17)$$

where

$$f(\mathbf{x}_i | \theta) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k), \quad (6.18)$$

N is the number of training vectors, $\theta = (\pi_1, \dots, \pi_K, \mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K)$ is the parameter vector, K is the number of mixtures, π_k is the k^{th} mixture weight, the mixture weights sum to unity ($\sum_{k=1}^K \pi_k = 1$), and $\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)$ is the Gaussian density for the k^{th} mixture with mean μ_k and covariance matrix Σ_k . Since no closed form solution exists for (6.17), $\hat{\theta}$ is found using the estimation-maximization (EM) algorithm [33].

The posterior probability of a song or artist given a mixture is calculated for all the mixtures in the UBM and are concatenated to form a vector to represent the song or artist. Note that the posterior probabilities are found using all the data from an artist or song. An alternate view of this procedure is that the final GMM for a particular song or artist is the maximum-likelihood solution when the means and covariances are fixed to the UBM. The spectral-based approach using the bag-of-timbres approximation in the PRank algorithm and the discriminative-training algorithm will be denoted as BoT/PRank and BoT/DTRank, respectively.

6.3.4 Spectral-based approach 3: GMM Supervectors

While the MeanVar approach models an artist or song with a single Gaussian, GMMs are a more common representation for MIR applications. However, converting a GMM to a vector cannot be accomplished by simply stacking mean vectors and unwrapped covariance matrices. The reason is shown in Figure 6.1. Each song is modeled with a GMM individually; therefore, there is no guarantee that mixtures will be labeled in a consistent fashion. In Figure 6.1, the mixtures are labeled such that the second mixture in Song B (m_{B2}) is closer to the first mixture in Song A (m_{A1}) than to the second mixture in Song A (m_{A2}). Therefore, the true distance between the songs is smaller than the distance found because the means are not stacked in the correct order:

$$\text{dist} \left(\begin{bmatrix} \mu_{11} \\ \mu_{12} \end{bmatrix}, \begin{bmatrix} \mu_{12} \\ \mu_{22} \end{bmatrix} \right) \geq \text{dist} \left(\begin{bmatrix} \mu_{11} \\ \mu_{12} \end{bmatrix}, \begin{bmatrix} \mu_{22} \\ \mu_{21} \end{bmatrix} \right) \quad (6.19)$$

where μ_{ij} is the mean of the j^{th} mixture in the i^{th} song and $\text{dist}(\cdot)$ is an appropriate distance metric, e.g., Euclidean. Finding the ordering of mixtures that produces the smallest distance between two GMMs is a computationally intensive task.

One solution to vectorizing GMMs is a successful approach to speaker identification: the supervector approach [23]. The rationale behind the supervector approach is shown in Figure 6.2. Like the BoT approach, the acoustic space is modeled using

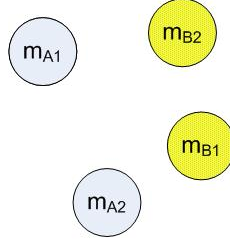


Figure 6.1: Illustration on the lack of calibration when using GMMs to model two objects individually. Blue, solid circles are mixtures for song A, and yellow, textured circles are mixtures for song B.

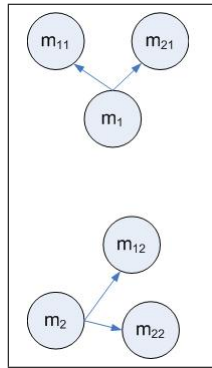


Figure 6.2: Rationale of the supervector approach. Each mixture of an artist or song GMM is an adapted mixture from a universal model.

all the training data to build a UBM. Using the mixtures in the UBM as a set of calibration points, a new GMM representing a particular song or artist is found using MAP adaptation [41]. The means of the adapted GMM are then concatenated to form a supervector. The steps to modeling an acoustic object (i.e., a song or artist) with a supervector is shown in Figure 6.3.

Following the estimation of the UBM, an artist-level or song-level density is estimated by MAP adaptation, which is based on Bayes theory and attempts to find the mode of the posteriori density:

$$\hat{\theta} = \arg \max_{\theta} P(\theta | \mathcal{X}_A) = \arg \max_{\theta} f(\mathcal{X}_A | \theta) P(\theta), \quad (6.20)$$

where \mathcal{X}_A is the training data specific to artist (or song) A . While any probability

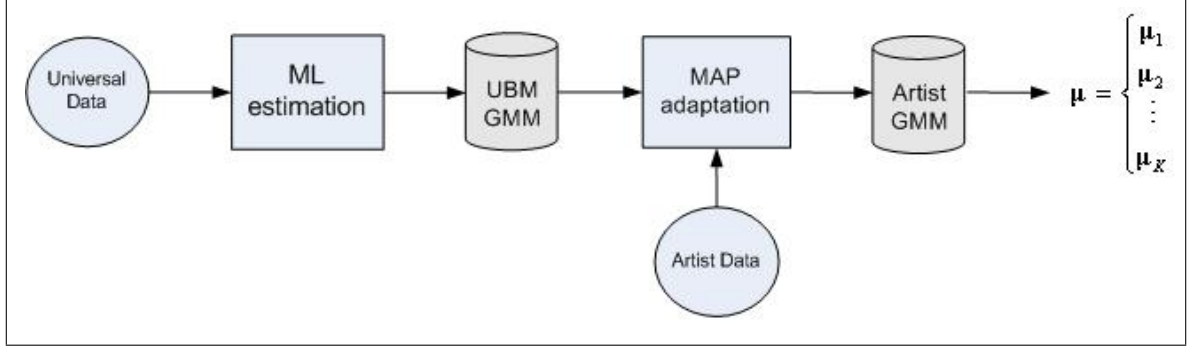


Figure 6.3: GMM-based supervector approach.

function may be used for the prior density, $P(\theta)$, in the absence of additional information a conjugate prior leads to a natural formulation [32]; however, no such density exists for a GMM. A solution is to view a GMM as a product of two densities [41]:

$$P(\theta) = P(\pi_1, \dots, \pi_K) \prod_{k=1}^K P(\mu_k, \Sigma_k). \quad (6.21)$$

The interpretation of (6.21) is the mixtures are chosen by a multinomial probability function; therefore, the natural conjugate prior, $P(\pi_1, \dots, \pi_K)$, is a Dirichlet density. Next, the individual mixture parameters are chosen by the conjugate prior for the multi-dimensional Gaussian with an unknown mean and variance: the normal-inverse-Wishart density [32]. Therefore the conjugate prior, $P(\mu_k, \Sigma_k)$, is a multi-dimensional Gaussian. Note that (6.21) also assumes independence between the mixture parameters and the Gaussian parameters. With these assumptions, a solution is found using the EM algorithm [41].

Following MAP adaptation, each artist-specific (or song-specific) density is converted to a vector by concatenation the means of the artist-specific density. Note that MAP adaptation provides a method of insuring that each vector is calibrated from the same source by using the mixtures in the UBM as starting points. The supervector approach using the PRank algorithm and the discriminative-training algorithm will be denoted as Super/PRank and Super/DTRank, respectively.

Table 6.1: Comparison of ratings distribution for the Amazon dataset in [140] and the Yahoo! dataset.

Rating	5	4	3	2	1
Amazon [140]	68.50%	18.70%	5.86%	2.71%	4.27%
Yahoo!	14.92%	13.07%	13.41%	11.44%	47.16%

6.4 Evaluation Metrics

The performance of the different system configurations will be evaluated using three performance metrics: ARL, average classification error (ACE), and normalized discounted cumulative gain (NDCG). ARL is the objective in the original presentation of PRank [29] and is given in (2.6). ACE gives the average number of test samples that are misclassified to have a different rank and does not differentiate the degree of error. While improvement in these two statistics ultimately leads to good results, a system would still perform well as long as the songs are ranked in the correct order. That is, a good system is one that lists the most relevant or most liked songs at the top of the list of returned results. Further, ARL and ACE are less robust error measures when the number of test samples is not distributed across the rankings uniformly.

In Table 6.1, the distribution of ratings is shown for the Yahoo! dataset (see Section 6.5) and the database from Amazon in [140]. Both datasets are relatively unbalanced, but in different ways. While the Amazon dataset prefers higher ratings, the Yahoo! dataset prefers lower ratings. A possible explanation for this difference lies in the nature of the two datasets. The ratings from Amazon are from customer reviews and require a user to visit Amazon to post a review after a purchase. Users are more likely to put forth this effort if their reaction is strong. Further, users are likely to only purchase items that they know are likely to be enjoyed. In contrast, the ratings from Yahoo! are from a radio website that presents material to the user. Therefore, there is a stronger incentive to rate music that is hated, e.g., to prevent similar music from being played in the future.

Regardless of the nature of the ratings, such an unbalanced dataset means that a naive classifier that only outputs the most common rating can perform well. Therefore, another evaluation measure is NDCG, which returns a ratio between a given system and a system that returns the results perfectly. Further, the importance of a prediction declines as one moves down the list, so more weight is given to items appearing at the top of the list. To find the NDCG, the predictions scores for Prank and DTRank are sorted from highest to lowest. Note that the prediction score for a test vector, \mathbf{x} , is the inner product between \mathbf{x} and the hyperplane \mathbf{w} in (2.5). Once sorted, the discounted cumulative gain (DCG) at level k is

$$\text{DCG}(k) = \sum_i^k \frac{2^{y_i-1}}{\log(i+2)}. \quad (6.22)$$

Note that y_i in (6.22) is the true rank for the document returned in position i . Because the DCG varies with the number of possible returned results (i.e., the number of possible test samples), it is common to normalize the DCG by the best possible DCG:

$$\text{NDCG}(k) = \frac{\text{DCG}(k)}{\hat{\text{DCG}}(k)}, \quad (6.23)$$

where $\hat{\text{DCG}}$ is the DCG when the results are returned perfectly.

6.5 Experiment 1: Explicit Artist Prediction

The first experiment uses explicit artist rankings from the Yahoo! Music User Ratings of Musical Artists database [139], which consists of over 11 million ratings of 98,211 artists by almost two million anonymous users. The scores are integer values between 0 and 100, except for a special value of 255 that indicates the artist is never to be played again. The scores are mapped to ranks from one to five by setting the rank equal to the quotient when dividing by 20. The special value of 255 is mapped to a rank of 1. A higher rank means the item is more preferred than a lower rank.

Table 6.2: Results for different system configurations with 250 training vectors. See text for details.

System	ARL	ACE	NDCG(5)	NDCG(10)	NDCG(15)
ASM/DTrank(1000/10)	0.8901	0.4719	0.5452	0.5994	0.6733
ASM/PRank	1.1976	0.5596	0.4458	0.4840	0.5303
MeanVar/DTrank	1.2024	0.5329	0.4838	0.5357	0.5827
MeanVar/PRank	1.3109	0.5551	0.4176	0.4621	0.5116
MAP/DTRank	1.1785	0.5281	0.5199	0.5847	0.6535
MAP/PRank	1.2027	0.5583	0.4284	0.4853	0.5327
Anchor/DTRank	1.1776	0.5297	0.4941	0.5374	0.5833
Anchor/PRank	1.2294	0.5776	0.3913	0.4519	0.5084

Since this dataset does not contain audio, the acoustic data is provided by the USPop dataset [14], which consists of the first 20 MFCCs, including the zeroth coefficient, from 8,764 songs and 400 artists. The final format of the acoustic data is first 13 coefficients, including the zeroth coefficient, which are appended with the first and second order derivatives to produce a 39-dimensional vector. Further, cepstral mean and variance normalization are performed for all systems except for the MeanVar systems, which performed best without cepstral mean and variance normalization.

The resulting training and test databases are provided by the intersection of artists from the two datasets. Further, users were retained if they rated at least 300 artists and rated at least five artists at each rank. The first 50 users serve as a cross-validation set to find the optimal number of mixtures in the BoT and Super systems, the number of singular values in the ASM system, and the parameter α .

The results for the different system configurations system configurations are shown in Table 6.2. The ASM systems performed better than the GMM systems when using the same classifier. This demonstrates that user preferences are better modeled by incorporating dynamic information, including the syntactical arrangement of sounds. Further, the discriminative-training systems perform better than the systems that use PRank to perform the final ranking. In fact, the ASM/DTRank system is within a single rank on average and performs the best for all the evaluation measures.

Table 6.3: Results the ASM/DTRANK system using different training sizes.

Training Size	ARL	ACE	NDCG(5)	NDCG(10)	NDCG(15)
250	0.9321	0.4792	0.5365	0.6018	0.6721
200	0.9285	0.4753	0.5313	0.5930	0.6609
150	0.9155	0.4807	0.5190	0.5760	0.6505
100	0.9079	0.4849	0.4875	0.5505	0.6200
50	0.9066	0.4977	0.4592	0.5112	0.5836

The performance of the ASM/DTrank system using a different number of training samples is shown in Table 6.3. The training samples are selected randomly from the original set of 250. As the number of training samples decrease, the systems perform poorer in terms of ACE and NDCG. Note that the ARL decreases, which is not the expected result. Closer analysis revealed this had to do with the unbalanced dataset. For many users, every artist was predicted to have a rank of one when there were fewer training samples. This resulted in an increase in ARL, even though ACE increased. However, because the DTRank algorithm focuses on minimizing misclassification, both ACE and NDCG still perform well when there is sufficient training data. When the number of training samples decreases, the number of misclassified test samples increase, which also causes the NDCG to decrease because the returned list of songs is not sorted as accurately. In fact, using only 50 training samples results in a very decreased NDCG, and using only 25 samples caused the algorithm to crash for some users because they lacked enough samples at each level.

One reason for the improved performance of the discriminative-training ordinal regression algorithm is shown in Figure 6.4, which shows the estimated empirical ranking loss in (6.1) against the true ranking loss in (2.6) for the first ten users. While, both the estimated and true ranking loss decrease as the number of iterations increases, the estimated empirical loss is greater than the true ranking loss. The true ranking loss is a discrete metric (either zero or one) whereas the estimated ranking loss is a real value bounded between zero and one. This means that for a misclassified

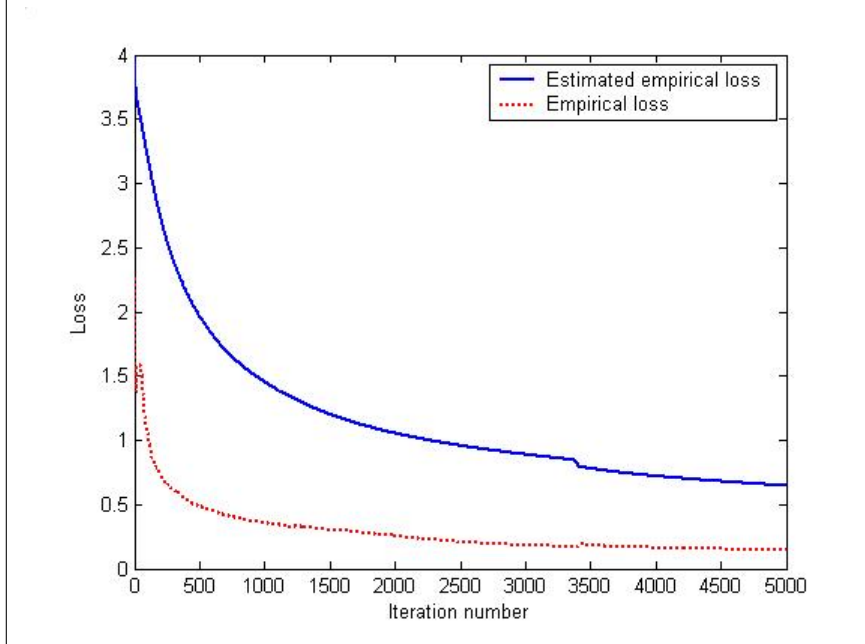


Figure 6.4: Estimated empirical ranking loss in (6.1) and true ranking loss in (2.6) averaged over the first ten users.

sample, \mathbf{x}_{mc} , that

$$L_{AR}(\mathbf{x}_{mc}; \mathbf{w}, \mathbf{b}) = 1 \geq L_{est}(\mathbf{x}_{mc}; \mathbf{w}, \mathbf{b}). \quad (6.24)$$

Therefore, the estimated training error also penalizes training samples that are classified correctly. This means that correctly classified training samples also contribute to the updates for \mathbf{w} and \mathbf{b} . In fact, it is seen in (6.9), (6.10), (6.12), and (6.13) that if the loss sigmoid is close to zero or one, then the update is small in magnitude. That is, samples closest to the decision boundary lead to the largest updates, which is similar to the effect of support vectors.

Further, it is seen in (6.9) and (6.10) that if the loss sigmoid is close to zero or one, the update is small in magnitude. This is shown for a typical user in Figure 6.5. A couple observations can be made from the figure, which shows the estimated loss for the decision boundary between $r = 3$ and $r = 4$ for each training sample at the first and last iteration. First, most training samples have errors that decrease significantly. Second, for the few remaining training samples that do not have a reduced error, the

error values are close to one and do not contribute to the update significantly. In effect, these outliers are ignored by the training algorithm, which has been shown to be an important quality of MCE [143]. Note that outliers in SVMs are not ignored and maximally contribute to the solution [138].

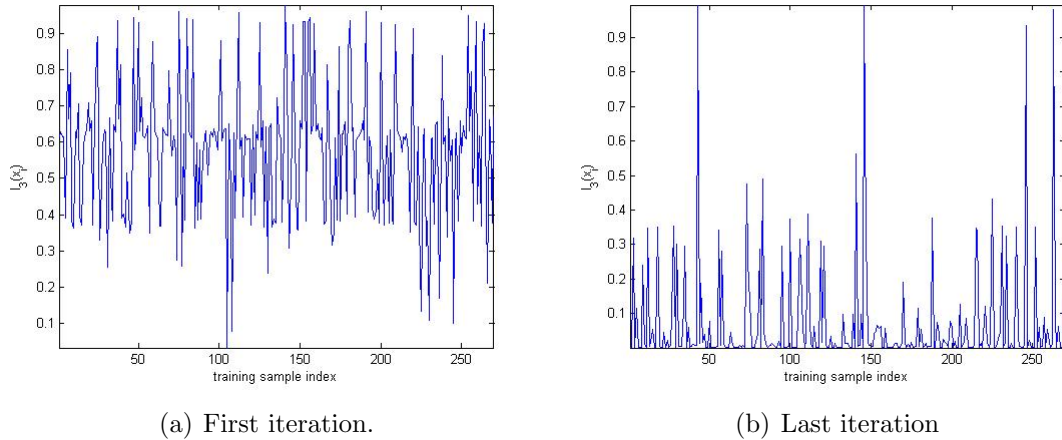


Figure 6.5: Values of the 0-1 loss approximation in (6.4) at the boundary between $r = 3$ and $r = 4$ for each training sample of a typical user at the first and last iteration.

The discriminative-training approach increases the margin of the decision boundary by penalizing both correctly and incorrectly misclassified samples. This is shown in Figure 6.6, which shows the misclassification function in (6.5) for a random user at iterations 10 and 2000. Recall that values that are more negative indicate a particular training example is classified correctly and with more confidence. As shown in the figure, the histogram of the misclassification function becomes more negative after the discriminative-training procedure has terminated, indicating that training examples are classified correctly more often and with higher certainty.

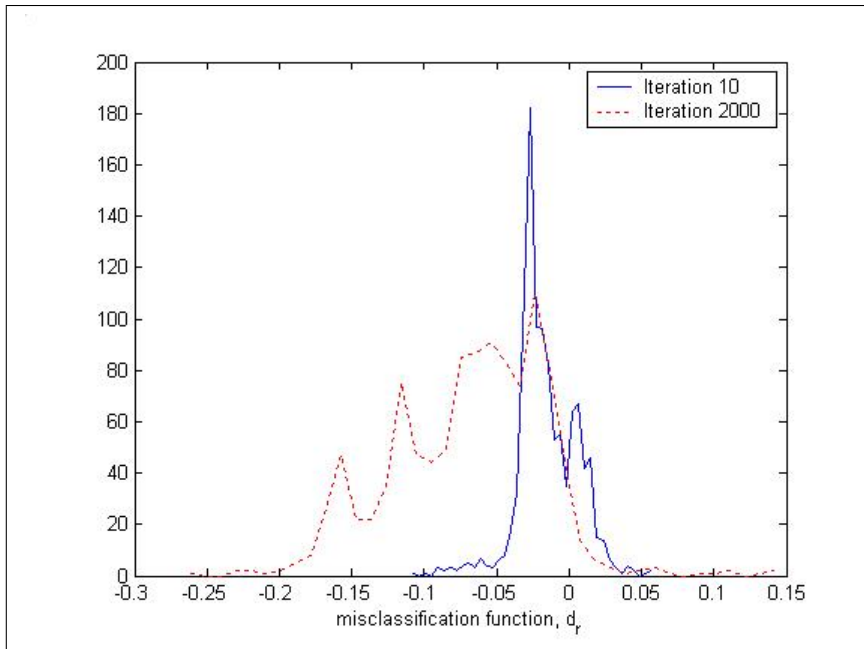


Figure 6.6: Misclassification measure given in (6.5) for iterations 10 and 2000 for a random user.

6.6 *Experiment 2: Implicit Song Rankings*

The explicit ratings in Section 6.5 require a user to label a sufficient number of artists or song. However, users are less likely to continue to use a system if they cannot see an immediate benefit to their effort. This section investigates the ratings prediction algorithms on implicit ratings gathered from a user’s listening history. Unlike most forms of multimedia (e.g., movies), a piece of music is often consumed more than once, and a user is likely to listen to a favorite song several times.

User listening histories for 1,000 users were gathered in [24] using the last.fm API¹. Only songs that matched the USPop2002 dataset are retained for the experiment in this section. Next, the complementary cumulative distribution for each user is calculated. Finally, ratings are assigned such that songs in the top 80-100% of the distribution receive a rating of 5, songs in the 60-80% quintile receive a rating of 4, etc. Users are retained if there are at least 5 songs at each rating. The final

¹www.last.fm/api

dataset is relatively small (100 users); therefore, no cross-validation is performed and parameters are the same as in Section 6.5.

The results in Table 6.4 demonstrate that predicting implicit ratings is a harder task. While the acoustic segment modeling approach still performs best in terms of ACE and NDCG, the performance with the supervector approach has a comparable ARL. There are several potential causes for the decrease in ratings prediction for the implicit versus explicit task. First, the mappings assumed that the cumulative distribution should be divided into bins that are equally spaced; however, the number of plays versus artist rank is distributed exponentially [24]. It is still unclear how the number of times a user listens to a song indicates a perceived level of likability. While it is likely that a user listens to a favorite song more than a hated song, it is not clear what the ratio is of the number of times a “5” is heard over a “4.” Second, the database does not account for the age of the song or the time when the user was first introduced to the song. An older song that is equally liked as a newer song is likely to be heard more often, but at similar rates; therefore, a more reliable measure may be frequency. However, this data is hard to collect on a large scale because most sites do not publish this information. One exception is last.fm; however, this requires tracking users over several weeks. Finally, the nature of the implicit ratings contains significance. Online radio stations like last.fm keep a record of songs that are presented to the listener on the website and, optionally, on the user’s computer or MP3 device. There is probably a higher value in songs that are explicitly selected for play versus songs that are presented to the user. In particular, many users listen to music while performing other tasks and their attention is divided [49]. Potentially, better databases can be obtained by leveraging user studies, which has been shown to help algorithmic development for web-based document retrieval [53].

Table 6.4: Results for ratings prediction on implicit ratings

System	ARL	ACE	NDCG(5)	NDCG(10)	NDCG(15)
ASM/DTrank	1.1569	0.6308	0.2821	0.3419	0.4234
MeanVar/DTrank	1.3881	0.6512	0.2565	0.3218	0.4009
MAP/DTRank	1.1533	0.6806	0.2524	0.3340	0.4180
Anchor/DTRank	1.4171	0.6731	0.2219	0.3029	0.3898

6.7 Summary

This chapter presents a novel ordinal regression algorithm for the purposes of predicting user ratings of musical items. The proposed algorithm is based on the discriminative-training technique of MCE training and is related to the MFoM classifier. The motivation behind the proposed algorithm is to minimize the ARL by viewing the ordinal regression problem as a multi-class classification problem with a coupled hyperplane. Like MCE, the proposed algorithm minimizes the number of misclassified samples with a smoothed approximation for the misclassification rate. Specifically, at each rank boundary, the 0-1 loss is approximated by a smooth and differentiable sigmoid. Substituting the sigmoid function into the 0-1 loss results in an approximation for the ARL that is optimized using GPD. Further, unlike conservative algorithms such as Prank, the proposed algorithm uses correctly classified samples in addition to misclassified samples to further optimize the objective. Finally, the classifier shares a similar property of MCE: immunity to outliers.

This proposed algorithm is compared to PRank using four representations of an artist or song: the acoustic segment modeling approach and three spectral-based approaches. The first spectral-based approach models an artist or song using a Gaussian distribution and then concatenates the mean vector and unwrapped covariance matrix into a single vector representation. The second spectral-based approach is based on the “bag-of-timbres” model and uses all the acoustic data to develop a GMM UBM. An artist or song is represented by a vector containing the posterior probabilities of

each mixture in the UBM. The third spectral-based approach is based on the successful supervector approach to automatic speaker identification. Each artist or song is modeled with a GMM, which is adapted from a UBM using MAP adaptation. Next, the means are concatenated to form a vector to represent an artist or song.

It is shown that the proposed ranking algorithm is superior to PRank, regardless of the data representation. Further, the acoustic segment modeling approach performs better than the spectral-based approaches on a set of explicit user ratings. On a set of implicit user ratings gathered from listening histories, the supervector approach is comparable to acoustic segment modeling in terms of ARL; however, the acoustic segment modeling approach is shown to rank highly rated songs higher in the list of returned results.

Overall, the implicit ratings are more difficult to optimize; however, the approach taken to map user listening history to preference ratings is ad-hoc. Like web documents, degrees of relevance could potentially be mined for additional information; however, more research is needed. For example, unlike web documents, a user may consume music passively.

However, By utilizing preference rankings, many of the issues of previous taxonomies for MIR are resolved. First, unlike traditional categories such as genre, preference rating personalizes each classifier and does not require a user to be educated on the makeup of each individual class. Further, categorical algorithms, such as genre, tags, mood, etc. assume that a positively labeled example implies a user will enjoy the material.

CHAPTER VII

IMPROVEMENTS AND EXTENSIONS

This chapter extends the system in Chapter 6, where a novel ordinal regression algorithm based on MCE training [54] is presented. While it is noted that using acoustic segment modeling as a front-end captures more temporal information than common “bag-of-frame” classifiers, the feature set is quite limited. The features used in Chapter 6 are MFCCs and describe the rough spectral shape of the signal (see Chapter 2). While MFCCs have been successful in ASR and MIR technologies, MFCCs are ill-suited for certain MIR tasks. For example, chord recognition does not perform well with MFCCs. However, multiple feature types can highlight different aspects of the signal.

Further, it is noted in Chapter 5 that notions of music similarity personal are partly influenced by social factors. Therefore, attempts have been made to create hybrid systems that have aspects of content-based systems and collaborative-filtering systems. While a few attempts have been made in the field of MIR (see Section 2.1.3), many attempts have been made to produce hybrid systems in document retrieval and movie recommendation.

This chapter demonstrates how the ASM/DTrank system can be easily extended to include multiple sources of information. In Section 7.1, an additional feature is added to incorporate harmonic information into the signal through the use of pitch class profiles. Further, Section 7.2 investigates how content-based systems impact the performance of hybrid systems.

7.1 *Incorporating Multiple Acoustic Features*

Previous studies have shown that PCP features help with the genre recognition problem when combined with MFCCs in “bag-of-frames” classifiers [127]. However, since “bag-of-frame” classifiers largely ignore temporal information, only aspects of key or the spread of pitch classes can be described. More explicit modeling of temporal factors are needed to model other musical factors. For example, melody is a form of musical syntax and can be described by a sequence of an underlying vocabulary. Research investigating the utility of modeling chord sequences in music is limited. One exception is [94], where it is found that chord n -gram sequences are able to distinguish genres; however, the number of genres is quite small (three) and the results are limited (61.1% accuracy). One reason for the limited results is the granularity of the model, which contained 24 chords (the 12 major and 12 minor triads). As noted in Section 2.1.2.3, such a coarse representation may not be adequate to define similarity for most MIR applications.

However, building a separate chord model for each possible extension, inversion, etc. is computationally prohibitive. Most chords will occur very rarely in the training data, which makes model estimation difficult. The acoustic segment modeling procedure is shown to capture harmonic information in Section 4.2 and performs almost as well as the baseline maximum likelihood procedure on a chord detection task. This section investigates whether incorporating harmonic information will improve the user rating prediction task in Chapter 6.

7.1.1 **Extending ASM to Harmonic Information**

The baseline algorithm uses the acoustic segment modeling procedure on MFCCs for the front-end and the MCE-based ranking algorithm for the backend. This algorithm is extended to include harmonic information by producing two channels that produce a pair ASM transcriptions for each song: one on MFCCs and one on PCPs. The

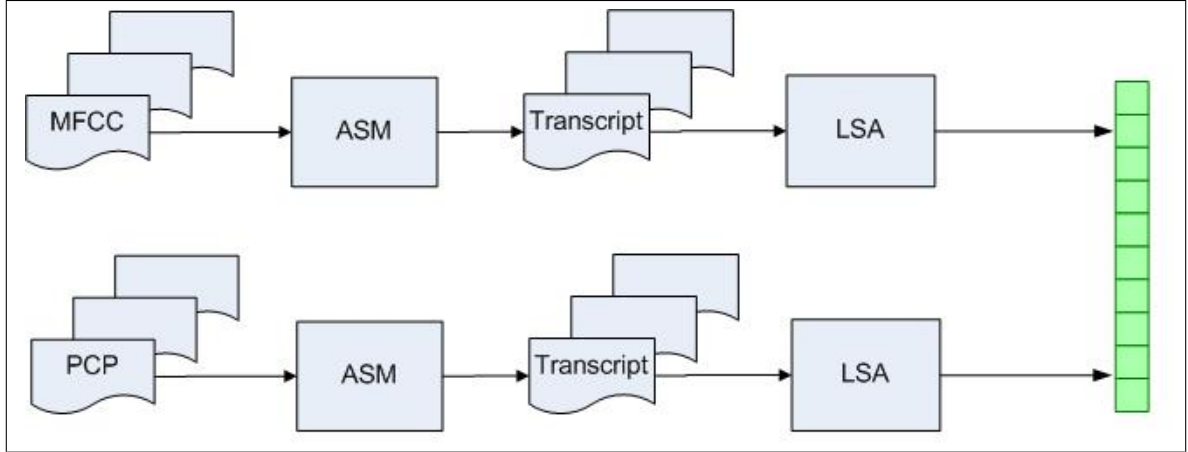


Figure 7.1: Front-end for combined MFCC/PCP approach.

system diagram is shown in Figure 7.1. For the MFCC channel, the ASMs trained in Section 6.3.1 are used. There are 128 MFCC-based ASMs, each modeled by a three-emitting state HMM, where each state contains an eight-mixture GMM with a diagonal covariance. The PCP-based channel uses the ASMs trained in Section 4.2. There are 80 PCP-based ASMs, each modeled by a single-emitting state GMM. The PCPs are found using HPSS, and the DFT is used to remove correlation in the PCP vector; therefore, each GMM uses a diagonal covariance matrix in each of the eight mixtures. A transcription is made for each song and each channel using Viterbi decoding. LSA (see Section 3.4.2.1) is performed separately on each channel. The two channels are combined by concatenating the resulting vector from each channel. The concatenated vectors serve as the inputs into the MCE-based ranking algorithm.

7.1.2 Database

Because the USPop dataset only contains MFCCs this section uses a private dataset owned by the author. The dataset contains 1,254 songs containing music that can be classified using the following terms: rock, pop, indie rock, metal, hip-hop, rap, r&b, and country. The user rankings are from the Yahoo! Music User Ratings of Musical Artists database [139] and are mapped to the private dataset. The final dataset consists of 296 artists and 106 users who rated at least 200 artists. For each

Table 7.1: Results using MFCCs only, PCPs only, and MFCCs and PCPs.

System	ARL	ACE	NDCG(5)	NDCG(10)	NDCG(15)
MFCC	0.8135	0.4216	0.4546	0.5047	0.5491
PCP	0.8798	0.4314	0.3661	0.4280	0.4890
MFCC/PCP	0.7272	0.4057	0.4581	0.5021	0.5563

user, 150 artists are used for training and the remaining artists are used for testing. The combined system is compared to using the ASM weighted-vectors using either MFCCs only or PCPs only. The evaluation metrics are the ARL, ACE, NDCG(5), NDCG(10), and NDCG(15), which are described in Section 6.4.

7.1.3 Results

The results are shown in Table 7.1. The system using only MFCC does better than the system using only PCPs in every evaluation metric. This superiority of MFCCs over PCPs has been noted in many “bag-of-frames” approaches and other static classifiers [127][9]. However, the PCP system still performs relatively well in terms of classification rate. One reason is the improved temporal modeling. Approaches that produced a single PCP vector for each piece or use “bag-of-frames” modeling techniques are essentially providing an estimate for the musical key and the spread of pitch classes. However, it is unlikely that people prefer a particular key since only a small percentage of the population possess absolute pitch [123]. Unlike these approaches, acoustic segment modeling incorporates syntax, which translates into melody. In fact, many users on last.fm have used the tag *melody* to annotate at least one song, and there are many other tags that contain the word *melody*.

To further investigate if the PCP system is able to capture a few musical aspects of the signal better than the MFCC system, the tag annotation experiment in Section 4.1 is repeated using the private dataset. Due to the smaller size of the private dataset, only ten tags are found with enough positively labeled songs. The PCP system is compared to the MFCC system in Table 7.2. The PCP system does better than

Table 7.2: Tag annotation EER for the PCP-based and MFCC-based ASM systems.

Tag/Attribute	MFCC	PCP
major key tonality	34.09	32.16
subtle use of vocal harmony	33.17	37.17
vocal centric aesthetic	38.29	46.58
acoustic rhythm guitars	28.70	37.40
extensive vamping	42.22	48.82
minor key tonality	43.70	35.80
mild rhythmic syncopation	44.63	43.90
mixed acoustic and electric instrumentation	35.71	39.56
acoustic rhythm piano	31.56	40.00

the MFCC system for only three of the ten tags; however, two of the tags (*major key tonality* and *minor key tonality*) describe the harmonic content of the signal. Note that this is a different result from [9], where it was found that MFCCs always produced a better system than PCPs. The reason is that [9] uses a spectral approach (i.e., a GMM estimated by a mixture of hierarchies algorithm); therefore, the PCP system in [9] cannot describe aspects of melody. Even though only unigrams and bigrams are used, the ASM approach is able to capture more elements of melody.

7.2 *Hybrid Approach*

As noted in Section 2.1, collaborative-filtering systems have generally performed better than content-based approaches; however, collaborative-filtering systems are susceptible to problems due to sparsity. Sparsity arises in any large-scale system because no users are able to use or buy a majority of the items in the system. This leads to many zeros in the resulting user-item matrix and decreases the odds of finding similar users. In the extreme case, this results in the “cold-start” problem where the system is unable to find a user or item that has no ratings, e.g., a new item. Many attempts to combine collaborative-filtering systems and content-based systems have been attempted and this has received some interest in the MIR community

(see Section 2.1.3). However, little research has investigated how the performance of content-based systems impact hybrid systems. This section uses the hybrid system in [77], where the content-based system replaces missing values in the collaborative-filtering system. However, [77] did not perform an analysis on how the quality of content-based systems impacts the performance of the hybrid system. While more complex hybrid systems are possible, this falls outside the scope of this dissertation and is left for future work. Instead, this section serves to demonstrate how the quality of the different systems in Chapter 6 impacts hybrid systems.

7.2.1 Collaborative-Filtering Baseline

This section uses the simple user-driven, neighbor Pearson coefficient algorithm that has been shown to be effective for the similar task of movie ratings prediction [20]. This algorithm is a memory-based algorithm that predicts the rating a given user (called the “active” user, u_a) using ratings given by other users in the system. Specifically, the predicted rating that u_a will give item j is

$$r_{a,j} = \bar{u}_a + \frac{\sum_{i \in \mathcal{N}_a} \text{sim}(u_a, u_i) (u_{i,j} - \bar{u}_i)}{\sum_{i \in \mathcal{N}_a} |\text{sim}(u_a, u_i)|}, \quad (7.1)$$

where $u_{i,j}$ is the rating that user i gave to item j , \bar{u}_i is the average rating that user i assigned, and $\text{sim}(a, i)$ is the similarity weight between user the active user and user i . Generally, a neighborhood of users, \mathcal{N}_a , is determined by finding the users that have the highest weight in terms of magnitude.

While many user similarity functions exist, a very popular and successful similarity function is the Pearson correlation coefficient [20]:

$$\text{sim}(u_a, u_i) = \frac{\sum_{j \in \mathcal{I}_{a,i}} (u_{a,j} - \bar{u}_a) (u_{i,j} - \bar{u}_i)}{\sqrt{\sum_{j \in \mathcal{I}_{a,i}} (u_{a,j} - \bar{u}_a)^2 \sum_{j \in \mathcal{I}_{a,i}} (u_{i,j} - \bar{u}_i)^2}}, \quad (7.2)$$

where $\mathcal{I}_{a,i}$ is the intersection of the items voted by user a , \mathcal{I}_a , and user i , \mathcal{I}_i . One weakness of the Pearson correlation coefficient is that if two users share only a few co-occurring ratings, the ratings can be artificially high [47]. To reduce the similarity

weight of users who do not have many items in common, the Pearson correlation coefficient is scaled by the significance weighting factor, $sg_{a,i}$, which is 1 if u_a and u_i have co-rated 50 items and $|\mathcal{I}_{a,i}|/50$ if they have co-rated less than 50 items.

7.2.2 Content-Boosted Collaborative Filtering

Sparsity decreases the likelihood that two users that are actually similar will be found because they will have few co-rated items. This section explores how an improved content-boosted algorithm will yield better performance when combined with a collaborative-filtering system. The hybrid system in this section was first proposed in [77] for movie recommendation; however, the authors did not investigate how the quality of the content-based system impacted performance.

First, content-based ratings from the discriminative-training ratings prediction algorithm replace user-item pairs that are missing explicit ratings to form a “pseudo user-ratings vector.” In addition, because content-based systems rely on having enough data to build an accurate user model, correlations between u_a and u_i are multiplied by a hybrid correlation weight,

$$hw_{a,i} = hm_{a,i} + sg_{a,i}, \quad (7.3)$$

where $hm_{a,i}$ is the harmonic mean weighting factor:

$$hm_{a,i} = \frac{2 * m_a * m_i}{m_a + m_i}, \quad (7.4)$$

where m_i is 1 if the user has rated more than 50 items and $|\mathcal{I}_i|/50$ otherwise. Note that $hm_{a,i}$ is biased towards the user that has rated fewer items.

Besides using just the content-based ratings of other users in the system, the content-based ratings for the current user can be leveraged as an additional user. A self-weighting factor accounts for the poor content-based ratings that arise from

having only a few training samples:

$$sw_a = \begin{cases} \frac{|\mathcal{I}_a|}{50} \times max & |\mathcal{I}_a| < 50 \\ max & otherwise \end{cases} . \quad (7.5)$$

The final predictions for item j and u_a is

$$r_{a,j} = \bar{u}_a + \frac{sw_a (c_{a,j} - \bar{u}_a) + \sum_{i \in \mathcal{N}_a} hw_{a,i} \text{sim}(u_a, u_i) (u_{i,j} - \bar{u}_i)}{sw_a + \sum_{i \in \mathcal{N}_a} hw_{a,i} \text{sim}(u_a, u_i)}, \quad (7.6)$$

where $c_{a,j}$ is the content-based rating of item j for u_a . Note that Pearson similarity weight, the user ratings, and the average user ratings are based on the pseudo user-ratings vectors.

7.2.3 Database and Evaluation Metrics

The user ratings set is the same found in Section 6.5, which is found by finding users from the Yahoo! Music User Ratings of Musical Artists [139] that rated at least 300 artists from the USPop dataset [14]. A more realistic scenario would have users rate a varying degree of artists. Therefore, for each user, 50 ratings are chosen as the test set and a random number of the remaining ratings are retained for training the collaborative-filtering system, the content-based system, and the hybrid system. The content-based system consists of the discriminative-training ratings prediction algorithm using the acoustic segment modeling procedure (see Section 6.3.1), the mean-var system (see Section 6.3.2), and the MAP-derived supervectors (see Section 6.3.4). The collaborative filtering system outputs a real number, therefore, ACE cannot be used. Further, comparisons using ARL are unfair given that the content-based algorithm is constrained to output an integer. Therefore, only the NDCG at levels of 5, 10, and 15 are reported.

Table 7.3: Results of three hybrid systems compared to a purely collaborative-filtering (CF) approach

System	NDCG(5)	NDCG(10)	NDCG(15)
CF	0.6104	0.6664	0.7248
ASM	0.7156	0.7522	0.8035
MeanVar	0.6519	0.6826	0.7355
MAP	0.6878	0.7245	0.7749

7.2.4 Results

A comparison of the three systems is shown in Table 7.3. It is seen that better content-based performance directly impacts the improvement in the hybrid systems. Further, even the poorest content-based system, MeanVar, yielded better performance when combined with the collaborative-filtering system than the collaborative-filtering system alone. It should be noted that the user-item matrix in these experiments is quite dense. The reason for the dense matrix is the relatively low number of artists in the USPop Dataset (396). However, previous research has shown that sparsity affects collaborative-filtering systems negatively and merging with a content-based system improves performance [77][141][140]. Therefore, this result demonstrates that hybrid approaches are beneficial even in cases where sparsity is not relatively high. Future work will examine larger datasets, but where rankings are found implicitly. In addition, more complex hybrid systems will be examined.

7.3 Summary

This chapter extends the content-based preference ratings prediction algorithm in Chapter 6 in two ways. The first incorporates an additional feature that is meant to capture melodic aspects of music: PCPs. In Section 4.2 it is shown that the acoustic segment modeling procedure is able to perform at a level near the state-of-the-art, which demonstrates that acoustic segment modeling can capture the melodic information using an unsupervised process. This chapter demonstrated that while MFCCs

still perform better than PCP for ratings prediction using ASMs, the combined system of MFCCs and PCPs performed better than MFCCs alone. The reasons for this improved performance are further investigating by using the MFCC-based ASMs and the PCP-based ASMs to detect musical attributes in a similar experiment as Section 4.1. It is found that MFCCs generally perform better than PCPs; however, PCPs are better at detecting musical attributes dealing with melodic qualities. This result is different from similar experiments on tagging [9]. The reason for the difference in results is that ASMs are able to capture temporal qualities of music better than the spectral-based approach in [9].

The second improvement in this chapter is to combine the content-based system with a collaborative-filtering system to create hybrid approach. As noted in Chapter 5, it is unlikely that any content-based system will perform perfectly in predicting user preferences. The reason is because notions of music similarity are not entirely based on the acoustic attributes in the signal. Rather, a combination of acoustic, personal, social, and environmental factors influence personal notions of similarity. However, improvement in content-based systems can lead to improved performance when combined with collaborative-filtering systems. This chapter investigated the direct link between improvement in content-based systems and hybrid systems. The content-based system serves two purposes. First, it decreases sparsity by filling in missing values for users in the neighborhood of the active user. Second, the content-based score for the active user serves separate voter whose weight is determined by the number of items the active user has rated. Results demonstrate that better content-based systems result in a better hybrid system. While better hybrid systems exist, this is outside the scope of this dissertation. Rather, the role of this experiment is to demonstrate that research into improved content-based systems should continue.

CHAPTER VIII

CONCLUSION

The objective of this dissertation is to generate personalized music recommendations using content-based analysis of musical signals. In Chapter 1, it is noted that most approaches to MIR suffer from three main limitations:

1. most acoustic content-based recommendation algorithms use a “bag-of-frames” model, where it is assumed that songs contain a simplistic, global audio texture.
2. genre, style, mood, and authors are appropriate categories for machine-oriented recommendation
3. most acoustic content-based recommendation technologies directly link low-level features to higher-order categories such as “songs I like,” genre, style, etc.

The first issue is addressed through the use of acoustic segment modeling. In Chapter 3, the genre recognition problem is tackled with an unsupervised temporal modeling approach inspired by ASR technology. A given song is tokenized by a universal temporal model set, called acoustic segment models (ASMs). These ASMs have similar characteristics of phonemes in speech, but they are estimated in an unsupervised fashion. The results on the genre classification task indicated that the ASM approach performed better than existing GMM approaches. However, despite the improved ability to capture temporal information, error rates remained high.

This increased ability to capture temporal information is shown in Chapter 4 on two different tasks. The first task is tag annotation and retrieval, where the acoustic segment modeling approach is compared to the baseline spectral-based algorithm in [125]. It is shown that the acoustic segment modeling approach performs better than

the baseline for a majority of the tags. More importantly, the acoustic segment modeling approach has the most improvement over the baseline when the tags describe temporal aspects in music. The second task is chord recognition. In the field of MIR, automatic chord recognition most resembles ASR. Indeed, the state-of-the-art approach is nothing more than the maximum-likelihood approach of a small vocabulary system with features that describe harmonic content [128]. It is found that the acoustic segment modeling approach captured the harmonic content to within 5% of the state-of-the-art. Further, it is noted that current chord recognition tasks are solving a problem similar to speaker-dependent ASR; that is, both the training and testing data derive from the same artist. Potentially, there may be an application for the acoustic segment modeling approach when several artists are considered; however, this is under further investigation. Regardless, Chapter 4 demonstrates that acoustic segment modeling is able to capture more temporal information than current MIR approaches.

The second and third limitation are jointly considered in Chapter 5, which examines the source of the performance ceiling that has been noted for genre recognition in this dissertation and elsewhere [5][76][88]. Specifically, it is found that a group of musicologists do not associate genre labels and mid-level acoustic features consistently. This is even true when the genre labels are assigned by the musicologists; therefore, it is not due to differences in opinion of genre assignments [4]. The implication is that it is not possible to fully describe a person’s notion of genre using only acoustic features or at least, known acoustic features. Similarly, other studies have demonstrated that general notions of similarity are based on non-acoustical cues. It is more probable that genre taxonomies are another example of “folksonomy” and unlikely to have arisen based strictly on the content of the music. For example, a musically knowledgeable fan will often decry a once favorite artist because the artist “sold-out” or due to overplaying on traditional radio, i.e., the “earworm” effect [30].

For content-based classification and retrieval, it is vital to have a well-defined taxonomy for two reasons. First, it is impossible to compare two systems when even the ground-truth labels are subjective. Second, content-based systems rely on labeled training data. While labeling errors are possible in any scenario, the amount seen in MIR is very large.

A large portion of the subjective nature in the ground-truth labels arises because any judgment about musical similarity is dependent on the given person. Therefore, this dissertation proposes a novel ordinal regression algorithm in Chapter 6 to guide music recommendation. From a qualitative perspective, ordinal regression is similar to reviews, where a discrete score is often given, e.g., “four out of five stars,” “six out of ten,” or “two thumbs up.” The ordinal regression algorithm presented in this dissertation is based on MCE training [54] and the maximum figure-of-merit (MFoM) classifier [40]. MCE training is a discriminative-training approach well-known in the ASR community that treats the recognition problem as a classification problem. The MCE loss function has been shown to correspond to the Parzen estimate of the theoretical classification risk [75]. Further, it is shown in [143] that MCE training is a form of soft-margin estimation like that seen in support vector machines [131], but with the added advantage of outlier detection. Not only is the discriminative-training ordinal regression algorithm more successful than a similar, conservative ordinal regression algorithm on an artist ratings prediction task, but the acoustic segment modeling procedure is shown to be superior than three common spectral-based approaches.

This improvement is further seen by leveraging additional features that highlight other aspects of music beyond timbre. Specifically, it is demonstrated that PCPs perform better at detecting the presence of melodic features (e.g., *major key tonality*) than MFCCs when the acoustic segment modeling procedure is used. This finding is different than spectral-based approaches to musical attribute detection, which found

that PCP features did not perform as well as MFCCs for even tags dealing with melody. The reason for this difference is the ability of ASMs to describe the temporal nature of sound, especially the underlying music syntax.

Further, it is shown that superior content-based recommendations directly impact the performance of hybrid systems. Ultimately, the best performing music recommendation systems will leverage both acoustic, personal, environmental, and social factors to produce better recommendations. While content-based systems tend to produce recommendations that are more novel than collaborative-filtering systems [24], content-based systems cannot detect many social factors that affect a user's satisfaction with a song or artist. However, collaborative-filtering systems can make recommendations that are already known to the user. Further, collaborative-filtering systems suffer from the cold-start problem and sparsity. In addition, more research is needed to determine how to leverage both explicit and implicit user ratings. For example, the winning system for the Netflix prize leveraged the fact that user preferences drift over time, including the observed behavior that a user's ratings will concentrate around a single value when the user is rating multiple movies in the same session [58].

In the future, it is likely that using personal and environmental factors will improve music recommendation. Currently, people carry mobile devices that are well-connected and can potentially scan the environment to identify the appropriate music for the given setting. Better algorithms will one day lead to devices that are smart enough to detect the music we like, when we like it, and when it goes out of style.

REFERENCES

- [1] ANDERSON, C., *The Long Tail: Why the Future of Business is Selling Less of More*. New York, New York: Chris Anderson, 2006.
- [2] ATAL, B. S., “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification,” *The Journal of the Acoustical Society of America*, vol. 55, no. 6, pp. 1304–1312, 1974.
- [3] AUCOUTURIER, J.-J., DEFREVILLE, B., and PACHET, F., “The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music,” *Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.
- [4] AUCOUTURIER, J.-J. and PACHET, F., “Music similarity measures: What’s the use?,” in *Proceedings of the International Symposium for Music Information Retrieval*, pp. 157–163, 2002.
- [5] AUCOUTURIER, J.-J. and PACHET, F., “Improving timbre similarity: How high’s the sky?,” *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, pp. 1–18, 2004.
- [6] AUCOUTURIER, J.-J., PACHET, F., and SANDLER, M., “The way it sounds’: Timbre models for analysis and retrieval of music signals,” *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1028–1035, 2005.
- [7] AUCOUTURIER, J.-J. and PAMPALK, E., “Introduction-from genre to tags: a little epistemology of music information retrieval research,” *Journal of New Music Research*, vol. 37, no. 2, pp. 87–92, 2008.
- [8] BAHL, L. R., BROWN, P. F., DE SOUZA, P. V., and MERCER, R. L., “Maximum mutual information estimation of hidden Markov model parameters for speech recognition,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 49–52, 1986.
- [9] BARRINGTON, L., TURNBULL, D., YAZDANI, M., and LANCKRIET, G., “Combining audio content and social context for semantic music discovery,” in *ACM Special Interest Group on Information Retrieval*, 2009.
- [10] BARTSH, M. A. and WAKEFIELD, G. H., “To catch a chorus: Using chroma-based representations for audio thumbnailing,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 15–18, 2001.

- [11] BELLEGARDA, J., “Exploiting latent semantic information in statistical language modeling,” *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1279–1296, 2000.
- [12] BELLO, J. P. and PICKENS, J., “A robust mid-level representation for harmonic content in music signals,” in *Proceedings of the International Symposium for Music Information Retrieval*, pp. 183–189, 2005.
- [13] BENNETT, J. and LANNING, S., “The Netflix prize,” in *Proceedings of the KDD-Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 3–6, 2007.
- [14] BERENZWEIG, A., LOGAN, B., ELLIS, D. P. W., and WHITMAN, B., “A large-scale evaluation of acoustic and subjective music-similarity measures,” *Computer Music Journal*, vol. 28, no. 2, pp. 63–76, 2004.
- [15] BERGSTRA, J., CASAGRANDE, N., ERHAN, D., ECK, D., and KEGL, B., “Aggregate features and Adaboost for music classification,” *Machine Learning*, vol. 65, no. 2–3, pp. 473–484, 2006.
- [16] BERTIN-MAHIEUX, T., ECK, D., MAILLET, F., and LAMERE, P., “Autotagger: A model for predicting social tags from acoustic features on large music databases,” *Journal of New Music Research*, vol. 37, no. 2, pp. 115–135, 2008.
- [17] BISHOP, C. M., *Pattern Recognition and Machine Learning*. New York, New York: Springer Science, 2006.
- [18] BLINN, J. F., “What’s the deal with the DCT?,” *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 78–83, 1993.
- [19] BOLL, S. F., “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.
- [20] BREESE, J. S., HECKERMAN, D., and KADIE, C., “Empirical analysis of predictive algorithms for collaborative filtering,” in *Proceedings on Uncertainty in Artificial Intelligence*, July 1998.
- [21] BRONKHORST, A. W., “The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions,” *Acustica*, vol. 86, no. 1, pp. 117–128, 2000.
- [22] BROWN, J., “Calculation of a constant Q spectral transform,” *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.
- [23] CAMPBELL, W., STURIM, D., REYNOLDS, D. A., and SOLOMONOFF, A., “SVM based speaker verification using a GMM supervector kernel and NAP variability compensation,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 97–100, 2006.

- [24] CELMA, O., *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.
- [25] CELMA, O. and LAMERE, P., “Music recommendation tutorial,” in *Proceedings of the International Symposium for Music Information Retrieval*, 2007.
- [26] CHU, W. and KEERTHI, S. S., “New approaches to support vector ordinal regression,” in *Proceedings of the International Conference on Machine Learning*, pp. 145–652, 2005.
- [27] COLEMAN, M., *Playback: From the Victorrolato MP3, 100 Years of Music, Machines, and Money*. Cambridge, Massachusetts: Da Capo Press, 2003.
- [28] CORTES, C. and VAPNIK, V., “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [29] CRAMMER, K. and SINGER, Y., “Pranking with ranking,” in *Proceedings of the Neural Information Processing Systems Conference*, pp. 641–647, 2001.
- [30] CUNNINGHAM, S. J., DOWNIE, J. S., and BAINBRIDGE, D., ““the pain, the pain”: Modeling music information behavior and the songs we hate,” in *Proceedings of the International Symposium for Music Information Retrieval*, pp. 474–477, 2005.
- [31] DAVIS, S. B. and MERMELSTEIN, P., “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.
- [32] DEGROOT, M., *Optimal Statistical Decisions*. New York, New York: McGraw-Hill, 1970.
- [33] DEMPSTER, A. P., LAIRD, N. M., and RUBIN, D. B., “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.
- [34] DENG, L., ACERO, A., PLUMPE, M., and HUANG, X., “Large-vocabulary speech recognition under adverse acoustic environments,” in *Proceedings of the International Conference on Spoken Language Processing*, pp. 806–809, 2000.
- [35] DIETTERICH, T. G., “Approximate statistical tests for comparing supervised classification learning algorithms,” *Neural Computation*, vol. 10, pp. 1895–1923, 1998.
- [36] DUAN, K.-B. and KEERTHI, S. S., “Which is the best multiclass SVM method? an empirical study,” in *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pp. 278–285, 2005.

- [37] DUREY, A. S. and CLEMENTS, M. A., “Melody spotting using hidden Markov models,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2001.
- [38] FUJISHIMA, T., “Realtime chord recognition of musical sound: A system using Common Lisp Music,” in *Proceedings of the International Computer Music Conference*, pp. 464–467, 1999.
- [39] GALES, M. J. F., “Maximum likelihood linear transformations for HMM-based speech recognition,” tech. rep., Cambridge University, 1997.
- [40] GAO, S., WU, W., LEE, C.-H., and CHUA, T.-S., “A maximal figure-of-merit MFoM-learning approach to robust classifier design for text categorization,” *ACM Transactions on Information Systems*, vol. 24, no. 2, pp. 190–218, 2006.
- [41] GAUVAIN, J.-L. and LEE, C.-H., “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.
- [42] GJERDINGEN, R. and PERROTT, D., “Scanning the dial: the rapid recognition of music genres,” *Journal of New Music Research*, vol. 37, no. 2, pp. 93–100, 2008.
- [43] GOTO, M., HASHIGUCHI, H., NISHIMURA, T., and OKA, R., “RWC music database: Popular, classical, and jazz music databases,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2002.
- [44] GRACHTEN, M., SCHEDL, M., POHLE, T., and WIDMER, G., “The ISMIR cloud: a decade of ISMIR conferences at your fingertips,” in *Proceedings of the International Symposium for Music Information Retrieval*, 2009.
- [45] HARTE, C., SANDLER, M., ABDALLAH, S., and GÓMEZ, E., “Symbolic representation of musical chords: a proposed syntax for text annotations,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2005.
- [46] HAZEN, T. J. and ZUE, V. W., “Automatic language identification using a segment-based approach,” in *Proceedings of EUROASPEECH*, pp. 1307–1310, 1993.
- [47] HERLOCKER, J., KONSTAN, J., BORCHERS, A., and RIEDL, J., “An algorithmic framework for performing collaborative filtering,” in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 230–237, 1999.
- [48] HOMBURG, H., MIERSWA, I., MOLLER, B., MONK, K., and WURST, M., “A benchmark dataset for audio classification and clustering,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2005.

- [49] HU, X., DOWNIE, J. S., and EHMANN, A. F., “Exploiting recommended usage metadata: Exploratory analyses,” in *Proceedings of the International Symposium on Music Information Retrieval*, pp. 67–72, 2006.
- [50] HURON, D., *Sweet Anticipation: Music and the Psychology of Expectation*. Cambridge, Massachusetts: MIT Press, 2006.
- [51] JENNINGS, D., *Net, Blogs, and Rock ‘N’ Roll: How Digital Discovery Works and What it Means for Consumers, Creators, and Culture*. Boston, Massachusetts: Nicholas Brealey, 2007.
- [52] JOACHIMS, T., “Making large-scale SVM learning practical,” *Advances in Kernel Methods - Support Vector Learning*, 1999.
- [53] JOACHIMS, T., “Optimizing search engines using clickthrough data,” in *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pp. 133–142, 2002.
- [54] JUANG, B.-H., CHOU, W., and LEE, C.-H., “Minimum classification error rate methods for speech recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.
- [55] KATZ, S. M., “Estimation of probabilities from sparse data for the language model component of a speech recognizer,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, 1987.
- [56] KAWAKAMI, T., NAKAI, M., SHIMODAIRA, H., and SAGAYAMA, S., “Hidden Markov model applied to automatic harmonization of given melodies,” *IPSJ Technical Report*, vol. 1999-MUS-034, pp. 55–66, 2000. In Japanese.
- [57] KNESER, R. and NEY, H., “Improved backing-off for m -gram language modeling,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1995.
- [58] KOREN, Y., “Collaborative filtering with temporal dynamics,” in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pp. 447–456, 2009.
- [59] KOSTKA, S. and PAYNE, D., *Tonal Harmony with an Introduction to Twentieth-Century Music*. New York, New York: McGraw-Hill, fifth ed., 2004.
- [60] KOUTRIKA, G., EFFENDI, F. A., GYÖNGYI, Z., HEYMANN, P., and GARCIA-MOLINA, H., “Combating spam in tagging systems,” in *Proceedings of the Third International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, pp. 57–64, 2007.
- [61] KRISHNA, A. G. and SREENIVAS, T. V., “Music instrument recognition: From isolated notes to solo phrases,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 265–268, 2004.

- [62] KUO, H.-K. and LEE, C.-H., “Discriminative training of natural language call routers,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 1, pp. 24–35, 2003.
- [63] LAMERE, P., “Social tagging and music information retrieval,” *Journal of New Music Research*, vol. 37, no. 2, pp. 101–114, 2008.
- [64] LEE, K. and SLANEY, M., “Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 291–301, 2008.
- [65] LEGGETTER, C. and WOODLAND, P., “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech and Language*, no. 2, pp. 171–185, 1995.
- [66] LEGGETTER, C. J. and WOODLAND, P. C., “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.
- [67] LI, J., *Soft Margin Estimation for Automatic Speech Recognition*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, 2008.
- [68] LIDY, T., RAUBER, A., PERTUSA, A., and INTESA, J., “Combining audio and symbolic descriptors for music classification from audio,” in *Music Information Retrieval Information Exchange (MIREX)*, 2007.
- [69] LINDEN, G., SMITH, B., and YORK, J., “Amazon.com recommendations,” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [70] LOGAN, B., “Mel-frequency cepstral coefficients for music modeling,” in *Proceedings of the International Symposium for Music Information Retrieval*, 2000.
- [71] LOGAN, B. and SALOMON, A., “A music similarity function based on signal analysis,” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 745–748, 2001.
- [72] MACQUEEN, J. B., “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
- [73] MANDEL, M. I. and ELLIS, D. P. W., “Support vector machine active learning for music retrieval,” *Multimedia Systems*, vol. 12, no. 1, pp. 3–13, 2006.
- [74] MANNING, C. D. and SCHÜTZE, H., *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: MIT Press, 1999.
- [75] MCDERMOTT, E. and KATAGIRI, S., “A derivation of minimum classification error from the theoretical classification risk using Parzen estimation,” *Computer Speech and Language*, vol. 18, no. 2, pp. 107–122, 2004.

- [76] MCKAY, C. and FUJINAGA, I., “Music genre classification: is it worth pursuing and how can it be improved?,” in *Proceedings of the International Symposium on Music Information Retrieval*, pp. 101–106, 2006.
- [77] MELVILLE, P., MOONEY, R. J., and NAGARAJAN, R., “Content-based collaborative filtering for improved recommendations,” in *Proceedings of the National Conference on Artificial Intelligence*, pp. 187–192, 2002.
- [78] MENG, A., AHRENDT, P., LARSEN, J., and HANSEN, L. K., “Temporal feature integration for music genre classification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1654–1664, 2007.
- [79] MILLER, G. A., “WordNet: A lexical database for English,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [80] MURRAY, W., GILL, P., and WRIGHT, M., *Practical Optimization*. London, United Kingdom: Academic Press, 1981.
- [81] NAKAGAWA, S., UEDA, Y., and SEINO, T., “Speaker-independent, text-independent language identification by HMM,” in *Proceedings of the International Conference on Spoken Language Processing*, pp. 253–256, 1992.
- [82] NEY, H., “The used of a one-stage dynamic programming algorithm for connected word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 263–271, 1984.
- [83] NEY, H., HAEB-UMBACK, R., TRAN, B.-H., and OERDER, M., “Improvements in beam search for 1000-word continuous speech recognition,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1992.
- [84] NEY, H., MERGEL, D., NOLL, A., and PAESELER, A., “A data-driven organization of the dynamic programming beam search for continuous speech recognition,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1987.
- [85] ONO, N., MIYAMOTO, K., ROUX, J. L., KAMEOKA, H., and SAGAYAMA, S., “Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram,” in *Proceedings of the European Signal Processing Conference*, 2008.
- [86] ORTMANNS, S., NEY, H., and EIDEN, A., “Language-model look-ahead for large vocabulary speech recognition,” in *Proceedings of the International Conference on Spoken Language Processing*, pp. 2095–2098, 1996.
- [87] PACHET, F. and CAZALY, D., “A taxonomy for musical genres,” in *Proceedings of the RIAO Content-Based Multimedia Information Access Conference*, 2000.

- [88] PAMPALK, E., *Computational Models of Music Similarity and their Application in Music Information Retrieval*. PhD thesis, Vienna University of Technology, 2006.
- [89] PAMPALK, E., FLEXER, A., and WIDMER, G., “Improvements of audio-based music similarity and genre classification,” in *Proceedings of the International Symposium for Music Information Retrieval*, 2005.
- [90] PANAGAKIS, Y., KOTROPOULOS, C., and ARCE, G. R., “Music genre classification using locality preserving non-negative tensor factorization and sparse representations,” in *Proceedings of the International Society for Music Information Retrieval*, pp. 249–254, 2009.
- [91] PAPADOPOULOS, H. and PEETERS, G., “Large-scale study of chord estimation algorithms based on chroma representation and HMM,” in *Proceedings of the International Workshop on Content-based Multimedia Indexing*, 2007.
- [92] PAPOULIS, A. and PILLAI, S. U., *Probability, Random Variables, and Stochastic Processes*. New York, New York: McGraw-Hill, 2002.
- [93] PAUL, B., “Algorithms for an optimal A* search and linearizing the search in the stack decoder,” in *International Conference on Acoustics, Speech, and Signal Processing*, 1991.
- [94] PEREZ-SANCHO, C., RIZO, D., KERSTEN, S., and RAMIERZ, R., “Genre classification of music by tonal harmony,” in *International Workshop on Machine Learning and Music*, 2008.
- [95] PERROTT, D. and GJERDINGEN, R., “Scanning the dial: An exploration of factors in the identification of musical style,” in *Proceedings of the Society for Music Perception and Cognition Conference*, 1999.
- [96] POVEY, D. and WOODLAND, P., “Minimum phone error and I-smoothing for improved discriminative training,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 105–108, 2002.
- [97] QUATIERI, T. F., *Discrete-Time Speech Signal Processing: Principles and Practice*. Upper Saddle River, New Jersey: Prentice Hall, 2002.
- [98] RABINER, L. and JUANG, B.-H., *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.
- [99] RABINER, L. R., “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [100] RAPHAEL, C., “Automatic segmentation of acoustic musical signals using hidden Markov models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 360–370, 1999.

- [101] RAPHAEL, C., “Automatic transcriptions of piano music,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2002.
- [102] REED, J. and LEE, C.-H., “A study on music genre classification based on universal acoustic models,” in *Proceedings of the International Symposium on Music Information Retrieval*, pp. 89–94, 2006.
- [103] REED, J. T., UEDA, Y., SINISCALCHI, S., UCHIYAMA, Y., SAGAYAMA, S., and LEE, C.-H., “Minimum classification error training to improve isolated chord recognition,” in *Proceedings of the International Symposium for Music Information Retrieval*, pp. 609–614, 2009.
- [104] REN, J.-M., CHEN, Z.-S., and JANG, J.-S. R., “On the use of sequential patterns mining as temporal features for music genre classification,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2010.
- [105] REYNOLDS, D. A., QUATIERI, T. F., and DUNN, R. B., “Speaker verification using adapted Gaussian mixture models,” *Digital Signal Processing*, vol. 10, no. 1–3, 2000.
- [106] ROBSON, M., “How teenagers consume media,” tech. rep., Morgan Stanley Research Europe, 2009.
- [107] ROSENBERG, A. E., DELONG, J., LEE, C.-H., JUANG, B.-H., and SOONG, F. K., “The use of cohort normalized scores for speaker verification,” in *Proceedings of the International Conference on Spoken Language Processing*, pp. 599–602, 1992.
- [108] RUBNER, Y., TOMASI, C., and GUIBAS, L., “The Earth Mover’s Distance as a metric for image retrieval,” tech. rep., Stanford University, 1998.
- [109] SCARINGELLA, N. and ZOIA, G., “On the modeling of time information for automatic genre recognition systems in audio signals,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2005.
- [110] SCHEIN, A. I., POPESCU, A., UNGAR, L. H., and PENNOCK, D. M., “Methods and metrics for cold-start recommendation,” in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.
- [111] SCHÖLKOPF, B. and SMOLA, A. J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, Massachusetts: MIT Press, 2002.
- [112] SHARDANAND, U. and MAES, P., “Social information filtering: algorithms for automating “word of mouth”,” in *CHI 1995: SIGCHI Conference on Human Factors in Computing Systems*, 1995.

- [113] SHASHUA, A. and LEVIN, A., “Ranking with large margin principles: two approaches,” in *Neural Information Processing Systems Conference*, pp. 937–944, 2002.
- [114] SHEH, A. and ELLIS, D. P. W., “Chord segmentation and recognition using EM-trained hidden Markov models,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2003.
- [115] SHINODA, K. and LEE, C.-H., “A structural Bayes approach to speaker adaptation,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 276–287, 2001.
- [116] SLANEY, M., “Semantic-audio retrieval,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2002.
- [117] SOUNDSCAN, N., “State of the industry,” tech. rep., RIAA, 2008.
- [118] STENZEL, R. and KAMPS, T., “Improving content-based similarity measure by training a collaborative model,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2005.
- [119] STRANG, G., *Linear Algebra and its Applications*. Brooks Cole, 1988.
- [120] SUROWIECKI, J., *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. New York, New York: Doubleday, 2004.
- [121] SVENDSEN, T. and SOONG, F. K., “On the automatic segmentation of speech signals,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 77–80, 1987.
- [122] SYMEONIDIS, P., RUXANDA, M., NANOPOULOS, A., and MANOLOPOULOS, Y., “Ternary semantic analysis of social tags for personalized music recommendation,” in *Proceedings of the International Symposium for Music Information Retrieval*, pp. 219–224, 2008.
- [123] TAKEUCHI, A. H. and HULSE, S. H., “Absolute pitch,” *Psychological Bulletin*, vol. 113, no. 2, pp. 345–361, 1993.
- [124] TUCKER, R. C. F., CAREY, M. J., and PARRIS, E. S., “Automatic language identification using sub-word models,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 301–304, 1994.
- [125] TURNBULL, D., BARRINGTON, L., TORRES, D., and LANCKRIET, G., “Semantic annotation and retrieval of music and sound effects,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 467–476, 2008.

- [126] TURNBULL, D., LIU, R., BARRINGTON, L., and LANCHRIET, G., “A game-based approach for collecting semantic annotations of music,” in *Proceedings of the International Symposium on Music Information Retrieval*, pp. 535–539, 2007.
- [127] TZANETAKIS, G. and COOK, P., “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [128] UEDA, Y., UCHIYAMA, Y., NISHIMOTO, T., ONO, N., and SAGAYAMA, S., “HMM-based approach for automatic chord detection using refined acoustic features,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2010.
- [129] VAN DER MERWE, P., *Origins of Popular Style*. New York, New York: Oxford Claredon Press, 1989.
- [130] VAN DER WAL, T., “Folksonomy coinage and definition.” www.vanderwal.net/folksonomy.html.
- [131] VAPNIK, V., *The Nature of Statistical Learning Theory*. New York, New York: Springer-Verlag, 1995.
- [132] VASCONCELOS, N., “Image indexing with mixture hierarchies,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3–10, 2001.
- [133] WELLER, A., ELLIS, D. P. W., and JEBARA, T., “Structured prediction models for chord transcription of music audio,” in *Proceedings of the International Conference on Machine Learning and Applications*, pp. 590–595, 2009.
- [134] WEST, K. and COX, S., “Features and classifiers for the automatic classification of musical audio signals,” in *Proceedings of the International Symposium for Music Information Retrieval*, 2004.
- [135] WEST, K. and COX, S., “Finding an optimal segmentation for audio genre classification,” in *Proceedings of the International Symposium for Music Information Retrieval*, 2005.
- [136] WHITMAN, B., *Learning the Meaning of Music*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.
- [137] WILCOXON, F., “Individual comparisons by ranking methods,” *Biometrics*, pp. 80–83, 1945.
- [138] XU, L., CRAMMER, K., and SCHUURMANS, D., “Robust support vector machine training via convex outlier ablation,” in *Proceedings of the American Association for Artificial Intelligence*, pp. 536–542, 2006.

- [139] YAHOO!, “Yahoo! music user ratings of musical artists, version 1.0.” http://research.yahoo.com/Academic_Relations.
- [140] YOSHII, K. and GOTO, M., “Continuous PLSI and smoothing techniques for hybrid music recommendation,” in *Proceedings of the International Society for Music Information Retrieval*, pp. 339–344, 2009.
- [141] YOSHII, K., GOTO, M., KOMATANI, K., OGATA, T., and OKUNO, H. G., “Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences,” in *Proceedings of the International Symposium on Music Information Retrieval*, 2006.
- [142] YOUNG, S., OLLASON, D., VALTCHEV, V., , and WOODLAND, P., *The HTK Book (for HTK Version 3.2)*. Cambridge University Engineering Department, 2002.
- [143] YU, D., DENG, L., HE, X., and ACERO, A., “Large-margin minimum classification error training: a theoretical risk minimization perspective,” *Computer Speech and Language*, vol. 22, no. 4, pp. 415–429, 2008.
- [144] ZIPF, G., *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Cambridge, Massachusetts: Addison-Wesley, 1949.
- [145] ZISSMAN, M. A., “Comparison of four approaches to automatic language identification of telephone speech,” *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31–44, 1997.