

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Design and Development of a Semantic Music Discovery Engine

A Dissertation submitted in partial satisfaction of the requirements for the degree
Doctor of Philosophy

in

Computer Science and Engineering

by

Douglas Ross Turnbull

Committee in charge:

Professor Charles Elkan, Co-Chair
Professor Gert Lanckriet, Co-Chair
Professor Serge Belongie
Professor Sanjoy Dasgupta
Professor Shlomo Dubnov
Professor Lawrence Saul

2008

Copyright
Douglas Ross Turnbull, 2008
All rights reserved.

The Dissertation of Douglas Ross Turnbull is approved, and it is acceptable in quality and form for publication on microfilm:

Co-Chair

Co-Chair

University of California, San Diego

2008

DEDICATION

The dissertation is dedicated to my parents, Martha and Bruce Turnbull, who have always ensured that I receive a well-rounded and thorough education. They have provided me with innumerable learning opportunities and taught me important lessons about open-mindedness, creativity, dedication, humility, work ethic, thoughtfulness, balance, appreciation, understanding, and perspective.

This dissertation is also dedicated to my wife Megan Galbreath Turnbull, whose encouragement and support have been boundless. She continually humbles me with her willingness to help others.

EPIGRAPH

”Writing about music is like dancing about architecture - it’s a really stupid thing to want to do.” — Elvis Costello and others ¹

¹The exact origins of this quote continue to be the subject of debate. Other individuals who have been associated with it include Laurie Anderson, Steve Martin, Frank Zappa, Thelonious Monk, and Martin Mull.

TABLE OF CONTENTS

Signature Page	iii
Dedication	iv
Epigraph	v
Table of Contents	vi
List of Figures	ix
List of Tables	x
Acknowledgements	xii
Vita	xiv
Abstract of the Dissertation	xvii
Chapter 1. Semantic Music Discovery	1
1.1 The Age of Music Proliferation	2
1.1.1 Production	2
1.1.2 Distribution	3
1.1.3 Consumption	5
1.2 Music Search and Music Discovery	7
1.3 Semantic Music Discovery Engine Architecture	12
1.3.1 Information Collection	14
1.3.2 Information Extraction	15
1.3.3 Music Information Index	17
1.4 CAL Music Discovery Engine	17
1.5 Summary	20
Chapter 2. Using Computer Audition to Generate Tags for Music	22
2.1 Introduction	22
2.2 Related work	25
2.3 Semantic audio annotation and retrieval	28
2.3.1 Problem formulation	29
2.3.2 Annotation	30
2.3.3 Retrieval	32
2.4 Parameter Estimation	33
2.4.1 Direct Estimation	35
2.4.2 Model Averaging	36
2.4.3 Mixture Hierarchies	36

2.5 Semantically Labeled Music Data	38
2.5.1 Semantic Feature Representation	40
2.5.2 Music Feature Representation	40
2.6 Semantically Labeled Sound Effects Data	41
2.7 Model evaluation	42
2.7.1 Annotation	43
2.7.2 Retrieval	46
2.7.3 Multi-tag Retrieval	46
2.7.4 Comments	47
2.8 Discussion and Future Work	48
2.9 Acknowledgments	50
Chapter 3. Using a Game to Collect Tags for Music	56
3.1 Introduction	56
3.2 Collecting Music Annotations	58
3.3 The Listen Game	60
3.3.1 Description of Gameplay	61
3.3.2 Quality of Data	62
3.4 Supervised Multiclass Labeling Model	63
3.5 Evaluation of Listen Game Data	64
3.5.1 Cal500 and Listen250 Data	64
3.5.2 Qualitative Analysis	65
3.5.3 Qualitative Evaluation	67
3.5.4 Results	68
3.6 Discussion	69
3.7 Acknowledgments	70
Chapter 4. Comparing Approaches to Collecting Tags for Music	71
4.1 Introduction	71
4.2 Collecting Tags	72
4.2.1 Conducting a Survey	75
4.2.2 Harvesting Social Tags	76
4.2.3 Playing Annotation Games	76
4.2.4 Mining Web Documents	77
4.2.5 Autotagging Audio Content	78
4.3 Comparing Sources of Tags	78
4.3.1 Social Tags: Last.fm	81
4.3.2 Games: ListenGame	82
4.3.3 Web Documents: Weight-based Relevance Scoring	82
4.3.4 Autotagging: Supervised Multiclass Labeling	84
4.3.5 Summary	85
4.4 Acknowledgments	86

Chapter 5. Combining Multiple Data Sources for Semantic Music Discovery . . .	87
5.0.1 Related Work	88
5.1 Sources of Music Information	90
5.1.1 Representing Audio Content	90
5.1.2 Representing Social Context	91
5.2 Combining Multiple Source of Music Information	92
5.2.1 Calibrated Score Averaging	93
5.2.2 RankBoost	94
5.2.3 Kernel Combination SVM	95
5.3 Semantic Music Retrieval Experiments	96
5.3.1 Single Data Source Results	97
5.3.2 Multiple Data Source Results	99
5.4 Acknowledgments	99
Chapter 6. Concluding Remarks and Future Directions	100
6.1 Concluding Remarks	100
6.2 Future Directions	101
6.2.1 Academic Exploration	101
6.2.2 Commercial Development	104
Appendix A. Definition of Terms	108
Appendix B. Related Music Discovery Projects	111
B.1 Query-by-semantic-similarity for Audio Retrieval	111
B.2 Tag Vocabulary Selection using Sparse Canonical Component Analysis	112
B.3 Supervised Music Segmentation	112
References	113

LIST OF FIGURES

Figure 1.1: Architecture of the Semantic Music Discovery Engine:	13
Figure 1.2: CAL Music Discovery Engine: Main Page:	18
Figure 1.3: CAL Music Discovery Engine: advanced query (top) and results (bottom) for “Beatles folk ‘acoustic guitar’ calming”:	19
Figure 1.4: CAL Semantic Radio Player: displaying playlist for ‘aggressive rap’ query:	20
Figure 2.1: Semantic annotation and retrieval model diagram.:	29
Figure 2.2: Semantic multinomial distribution over all tags in our vocabulary for the Red Hot Chili Pepper’s “Give it Away”; 10 most probable tags are labeled.: .	30
Figure 2.3: Multinomial distributions over the vocabulary of musically-relevant tags. The top distribution represents the <i>query</i> multinomial for the three-tag query presented in Table 2.7. The next three distribution are the <i>semantic</i> multinomials for top three retrieved songs. :	34
Figure 2.4: (a) Direct, (b) naive averaging, and (c) mixture hierarchies parameter estimation. Solid arrows indicate that the distribution parameters are learned using standard EM. Dashed arrows indicate that the distribution is learned using mixture hierarchies EM. Solid lines indicate weighted averaging of track-level models. : .	35
Figure 3.1: Normal Round: players select the best word and worst word that describes the song.:	61
Figure 3.2: Freestyle Round: players enter a word that describes the song.:	61

LIST OF TABLES

Table 1.1: Summary of Music Information Index	17
Table 2.1: Automatic annotations generated using the audio content. Tags in bold are output by our system and then placed into a manually-constructed natural language template.	24
Table 2.2: Music retrieval examples. Each tag (in quotes) represents a text-based query taken from a semantic category (in parenthesis)	26
Table 2.3: Music annotation results. Track-level models have $K = 8$ mixture components, tag-level models have $R = 16$ mixture components. A = annotation length (determined by the user), $ \mathcal{V} $ = vocabulary size.	51
Table 2.4: Sound effects annotation results. $A = 6$, $ \mathcal{V} = 348$	52
Table 2.5: Music retrieval results. $ \mathcal{V} = 174$	52
Table 2.6: Sound effects retrieval results. $ \mathcal{V} = 348$	53
Table 2.7: Qualitative music retrieval results for our SML model. Results are shown for 1-, 2- and 3-tag queries.	54
Table 2.8: Music retrieval results for 1-, 2-, and 3-tag queries. See Table 2.3 for SML model parameters.	55
Table 3.1: Musical Madlibs: annotations generated directly using the semantic weights that are created by Listen Game, and automatically generated annotations where the song is presented to the Listen250 SML model as novel audio content.	66
Table 3.2: Model Evaluation: The semantic information for CAL models was collected using a survey, while the Listen model was train using data collected using Listen Game. We annotate each song with 8 words.	68
Table 4.1: Comparing the <i>costs</i> associated with five tag collection approaches: The bold font indicates a strength for an approach.	72
Table 4.2: Comparing the <i>quality</i> of the tags collected using five tag collection approaches: The bold font indicates a strength for an approach.	73
Table 4.3: Strengths and weaknesses of tag-based music annotation approaches	79
Table 4.4: Tag-based music retrieval: Each approach is compared using all <i>CAL500</i> songs and a subset of 87 more obscure <i>long tail</i> songs from the Magnatunes dataset. <i>Tag Density</i> represents the proportion of song-tag pairs that have a non-empty value. The four evaluation metrics (<i>AROC</i> , <i>Average Precision</i> , <i>R-Precision</i> , <i>Top-10 Precision</i>) are found by averaging over 109 tag queries. †Note that ListenGame is evaluated using half of the <i>CAL500</i> songs and that the results do not reflect the realistic effect of the popularity bias (see Section 4.3.2).	86

Table 5.1: Evaluation of semantic music retrieval. All reported ROC areas and MAP values are averages over a vocabulary of 95 tags, each of which has been averaged over 10-fold cross validation. The top four rows represent the individual data source performance. “Single Source Oracle” picks the best single source for retrieval given a tag, based on the test set performance. The final three approaches combine information from the four data sources using algorithms that are described in Section 5.2. Note the performance differences between single source and multiple source algorithms are significant (one-tailed, paired t-test over the vocabulary with $\alpha = 0.05$). However, the differences between between SSO, CSA, RB and KC are not statistically significant. 98

ACKNOWLEDGEMENTS

First, I would like to acknowledge Professor Gert Lanckriet for getting behind this research project having had little prior experience with the analysis of music. His guidance, encouragement and support have made this project flourish.

In addition, I'd like to acknowledge Professor Charles Elkan, Professor Sanjoy Dasgupta, and Professor Lawrence Saul for both their active roles developing my interests in machine learning and helping me see this dissertation through to completion. Professor Serge Belongie and Professor Nuno Vasconcelos have had a large influence on many of the signal processing and computer vision aspects of this work. Professor Shlomo Dubnov and Professor Miller Puckette have made a large impact on the music-related aspects of this projects. Lastly, Professor Gary Cottrell and Professor Virginia de Sa have supported this work in numerous ways over the various stages of development.

I'd also like the acknowledge Luke Barrington for his role as my research partner and co-author. Many of the ideas that are found within this dissertation were initially discussed during a break in a jam session with Luke (and Antoni Chan) in the fall of 2005. Without Luke's eternal optimism, creativity and hard work, this project would have been left on the shelf of undeveloped ideas. In addition, I like to thank my other co-authors (David Torres, Antoni Chan, Mehrdad Yazdani, Roy Liu) and other collaborators (Arshia Cont, Omer Lang, Brian McFee) in the Computer Audition Lab for their ideas and suggestions over the last few years. Lastly, I like to thank Damien O'Malley and Aron Tremble for helping me to look outside the walls of academia for inspiration.

Finally, I like to thank Professor Perry Cook and Professor George Tzanetakis, my undergraduate research advisors, for getting me started on research involving the analysis of music. More importantly, they taught me the importance of both having fun and being creative when conducting serious research. I'd also like to thank Doctor Masataka Goto and Doctor Elias Pampalk, my collaborators in Japan, for welcoming my ideas and challenging me with alternative perspectives.

Chapter 2, in part, is a reprint of material as it appears in the IEEE Transaction on

Audio, Speech, and Language Processing, Turnbull, Douglas; Barrington, Luke; Torres, David; Lanckriet, Gert, February 2008. In addition, Chapter 2, in part, is a reprint of material as it appears in the ACM Special Interest Group on Information Retrieval, Turnbull, Douglas; Barrington, Luke; Torres, David; Lanckriet, Gert, July 2007. The dissertation author was the primary investigator and author of these papers.

Chapter 3, in full, is a reprint of material as it appears in the International Conference on Music Information Retrieval, Turnbull, Douglas; Liu, Ruoran; Barrington, Luke; Lanckriet, Gert, September 2007. The dissertation author was the primary investigator and author of this papers.

Chapter 4, in full, is a reprint of material as it appears in International Conference on Music Information Retrieval, Turnbull, Douglas; Barrington, Luke; Lanckriet, Gert. September 2007. The dissertation author was the primary investigator and author of this papers.

Chapter 5, in full, is a reprint of an unpublished Computer Audition Laboratory technical report, Turnbull, Douglas; Barrington, Luke; Yazdani, Mehrdad; Lanckriet, Gert, June 2008. The dissertation author was the primary investigator and author of this papers with the exception of Section 5.2.3.

VITA

2008	Doctor of Philosophy, University of California, San Diego
2005	Master of Science, University of California, San Diego
2001	Bachelor of Science & Engineering, Princeton University

PUBLICATIONS

Published (Peer-Reviewed):

D. Turnbull, L. Barrington, and G. Lanckriet. **Five approaches to collecting tags for music.** *International Conference on Music Information Retrieval (ISMIR)*, 2008.

L. Barrington, M. Yazdani, D. Turnbull and G. Lanckriet. **Combining Feature Kernels for Semantic Music Retrieval.** *International Conference on Music Information Retrieval (ISMIR)*, 2008.

D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. **Semantic annotation and retrieval of music and sound effects.** *IEEE Transaction on Audio, Speech, and Language Processing*, February 2008.

D. Turnbull, G. Lanckriet, E. Pampalk, and M. Goto. **A supervised approach for detecting boundaries in music using difference features and boosting.** In *International Conference on Music Information Retrieval (ISMIR)*, 2007.

D. Turnbull, R. Liu, L. Barrington, D. Torres, and G. Lanckriet. **Using games to collect semantic information about music.** In *International Conference on Music Information Retrieval (ISMIR)*, 2007.

D. Torres, D. Turnbull, L. Barrington, and G. Lanckriet. **Identifying words that are musically meaningful.** In *International Conference on Music Information Retrieval (ISMIR)*, 2007.

D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. **Towards musical query-by-semantic description using the CAL500 data set.** In *ACM Special Interest Group on Information Retrieval (SIGIR)*, 2007.

L. Barrington, A. Chan, D. Turnbull, and G. Lanckriet. **Audio information retrieval using semantic similarity.** In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.

D. Turnbull, L. Barrington, and G. Lanckriet. **Modeling music and words using a multi-class naïve Bayes approach.** In *International Conference on Music Information Retrieval (ISMIR)*, 2006.

D. Turnbull and C. Elkan. **Fast recognition of musical genres using RBF networks.** *IEEE Transactions on Knowledge and Data Engineering*, 17(4), 2005.

Working Manuscript:

D. Turnbull, L. Barrington, M. Yazdani, and G. Lanckriet. **Combining Audio Content and Social Context for Semantic Music Discovery.** *Computer Audition Laboratory Technical Report*, 2008.

Patents (Pending):

D. Turnbull, R. Liu, L. Barrington, D. Torres, and G. Lanckriet. **Generating Audio Annotations for Search and Retrieval.** *U.S. Patent Application Number 12/052,299* March 20, 2008

FIELDS OF STUDY

Major Field: Computer Science

Studies in Computer Audition.

Professors Gert Lanckriet, Lawrence Saul and Shlomo Dubnov

Studies in Machine Learning and Data Mining.

Professors Gert Lanckriet, Charles Elkan, Lawrence Saul, Sanjoy Dasgupta, and Nuno Vasconcelos

Studies in Audio and Image Signal Processing

Professors Shlomo Dubnov, Serge Belongie, Nuno Vasconcelos, and Lawrence Saul

Studies in Multimedia Information Retrieval

Professors Gert Lanckriet and Nuno Vasconcelos

ABSTRACT OF THE DISSERTATION

Design and Development of a Semantic Music Discovery Engine

by

Douglas Ross Turnbull

Doctor of Philosophy in Computer Science and Engineering

University of California, San Diego, 2008

Professor Charles Elkan, Co-Chair

Professor Gert Lanckriet, Co-Chair

Technology is changing the way in which music is produced, distributed and consumed. An aspiring musician in West Africa with a basic desktop computer, an inexpensive microphone, and free audio editing software can record and produce reasonably high-quality music. She can post her songs on any number of musically-oriented social networks (e.g., MySpace, Last.fm, eMusic) making them accessible to the public. A music consumer in San Diego can then rapidly download her songs over a high-bandwidth Internet connection and store them on a 160-gigabyte personal MP3 player. As a result, millions of songs are now instantly available to millions of people. This ‘Age of Music Proliferation’ has created the need for novel music search and discovery technologies that move beyond the “query-by-artist-name” or “browse-by-genre” paradigms.

In this dissertation, we describe the architecture for a semantic music discovery engine. This engine uses information that is both *collected* from surveys, annotation games and music-related websites, and *extracted* through the analysis of audio signals

and web documents. Together, these five sources of data provide a rich representation that is based on both the audio content and social context of the music. We show how this representation can be used for various music discovery purposes with the *Computer Audition Lab (CAL) Music Discovery Engine* prototype. This web application provides a music *query-by-description* interface for music retrieval, recommends music based on acoustic similarity, and generates personalized radio stations.

The backbone of the discovery engine is an *autotagging system* that can both annotate novel audio tracks with semantically meaningful *tags* (i.e. a short text-based token) and retrieve relevant tracks from a database of unlabeled audio content given a text-based query. We consider the related tasks of content-based audio annotation and retrieval as one supervised multi-class, multi-label problem in which we model the joint probability of acoustic features and tags. For each tag in a vocabulary, we use an annotated corpus of songs to train a Gaussian mixture model (GMM) over an audio feature space. We estimate the parameters of the model using the weighted mixture hierarchies Expectation Maximization algorithm. When compared against standard parameter estimation techniques, this algorithm is more scalable and produces density estimates that result in better end performance. The quality of the music annotations produced by our system is comparable with the performance of humans on the same task. Our *query-by-semantic-description* system can retrieve appropriate songs for a large number of musically relevant tags. We also show that our audition system is general by learning a model that can annotate and retrieve sound effects.

We then present *Listen Game*, an online, multiplayer *music annotation game* that measures the semantic relationship between songs and tags. In the normal mode, a player sees a list of semantically related tags (e.g., genres, instruments, emotions, usages) and is asked to pick the best and worst tag to describe a song. In the freestyle mode, a user is asked to suggest a tag that describes the song. Each player receives real-time feedback (e.g., a score) that reflects the amount of agreement amongst all of the players. Using the data collected during a two-week pilot study, we show that we can effectively train our autotagging system.

We compare our autotagging system and annotation game with three other approaches to collecting tags for music (conducting a survey, harvesting social tags, and mining web documents). The comparison includes a discussion of both scalability (financial cost, human involvement, and computational resources) and quality (cold start problem, popularity bias, strong vs. weak labeling, tag vocabulary structure and size, and annotation accuracy). Each approach is evaluated using a tag-based music information retrieval task. Using this task, we are able to quantify the effect of popularity bias for each approach by making use of a subset of more popular (short head) songs and a set of less popular (long tail) songs.

Lastly, we explore three algorithms for combining semantic information about music from multiple data sources: RankBoost, kernel combination SVM, and a novel algorithm which is called Calibrated Score Averaging (CSA). CSA learns a non-parametric function that maps the output of each data source to a probability and then combines these probabilities. We demonstrate empirically that the combining of multiple sources is superior to any of the individual sources alone, when considering the task of tag-based retrieval. While the three combination algorithms perform equivalently on average, they each show superior performance for some of the tags in our vocabulary.

Chapter 1

Semantic Music Discovery

The music industry is going through a dynamic period: the big record companies are losing their grip as CD sales decline, handheld music devices create new markets around the legal (and illegal) downloading of music, social networks bring musicians and fans closer together than ever before, and music websites (e.g., Last.fm, Pandora, Rhapsody) provide endless streams of new and exciting music from around the world. As a result, *millions of people now have access to millions of songs.*

While this current ‘Age of Music Proliferation’ provides new opportunities for producers (e.g., artists) and consumers (e.g., fans), it also creates a need for novel music search and discovery technologies. In this dissertation, we describe one such technology, which we call a *semantic music discovery engine*, that is a flexible and natural alternative to existing technologies. We refer to our system as *discovery engine* (as opposed to a *search engine*) because it is designed to help users discover novel music, as well as uncover new connections between familiar songs and artists. The term *semantic* reflects the fact that our system is built around a *query-by-description* paradigm where users can search for music using a large, diverse set of musically-relevant concepts in a natural language setting. For example, our semantic music discovery engine enables a music consumer to find “mellow classic rock that sounds like the Beatles and features acoustic guitar.”

In this chapter, we will discuss how technology is changing the music industry and describe a number of existing techniques for finding music. This highlights the need for powerful new music search and discovery technologies. We will then present the architecture for our semantic music discovery engine and introduce the CAL Music Discovery Engine prototype. This functional prototype explores a number of music discovery tasks: using query-by-description for music retrieval, generating automatic music reviews, calculating semantic music similarity, and creating playlists for personalized Internet radio.

1.1 The Age of Music Proliferation

Technology is changing the way music is produced, distributed and consumed. An amateur musician with a laptop computer, a microphone, and free audio editing software can record and produce reasonably high-quality music. She can then post her songs on any number of music-oriented websites or social networks making them accessible to the general public. A music fan can then rapidly download her songs over his high-bandwidth Internet connection and store them on his 160-gigabyte personal MP3 player.

In the following subsections, we will discuss ways in which recent technological developments have created the problem of *connecting millions of people to millions of songs*. We will also comment on some of the many social, legal and economic aspects that are involved with the production, distribution and consumption of music.

1.1.1 Production

In the early 1990's, the compact disc (CD) replaced the cassette tape as the leading medium for music distribution due to its small size, high-fidelity digital format, lack of deterioration with playback, and improved functionality (e.g., skipping/replaying

songs)¹. At that time, there was a significant cost associated with producing an album: recording the music in a studio, mixing the raw audio on an expensive sound board, producing a digital master copy, and using the master to press each CD in a clean room. In order, to make this process financially profitable, an artist (or record label) would have to sell hundreds of thousands of CDs.

Today, the production pipeline has changed in many ways. First, almost every personal computer comes equipped with a sound card, an audio input for a microphone, an audio output for speakers or headphones, and a CD-burner. For a few hundred dollars, any bedroom, basement or garage is a potential recording studio. Second, physical multitrack mixing boards can be emulated using software. Popular audio editing software packages include Garageband which comes with every Apple computer, and Audacity which is downloaded over a million times per month [Stokes (2007)]. In addition, professional software packages, like ProTools, Adobe Audition and Logic Audio, have come down in price (and can be illegally obtained for free using file sharing sites). Third, relatively compact MP3 audio files, high-bandwidth Internet connections, and inexpensive hard disks have eliminated the need for the physical transport and storage of music using CDs.² As a result of these low production costs, an amateur musician has few barriers to entry in a industry that was once the exclusive domain of the big record companies.

1.1.2 Distribution

Just as inexpensive computer hardware and software has significantly affected music production, the Internet has affected the ways in which music is distributed. In the late 1990s, peer-to-peer (P2P) file sharing services became a popular way to (illegally) distribute music. The most famous P2P company is Napster which was launched in June of 1999. After peaking with 1.5 million simultaneous users, Napster was shut

¹http://en.wikipedia.org/wiki/Compact_Cassette#Decline (Accessed May 2008)

²Ironically, the MP3 standard was finalized in 1992, the same year that CDs first out sold cassette tapes in the United States Rosen (2000).

down in the summer of 2001 by a lawsuit over copyright infringement that was filed by the Recording Industry of America Association (RIAA). Numerous P2P networks have followed Napster and continue to flourish despite the constant threat of legal action from the recording industry.

Seeking to develop a legal system to sell downloadable music, Apple launched their iTunes music store in 2003. In order to meet the piracy protection concerns of the major record companies, Apple began selling songs that were protected by a digital rights management (DRM) copyright protection system. Much to the dislike of the consumer, DRM limited iTunes music to the Apple's iPod portable music player, placed limits on how many times the music could be copied onto different computers, and made it difficult to recode the music into other (non-DRM) formats. However, fueled by the strength of iPod and gift card sales, it was announced in April of 2008 that Apple iTunes was the largest music retailer in the United States and hosted the largest catalog of downloadable music tracks (6+ million) on the Internet [Kaplan (2008)]. As a result of Apple's success, numerous companies, including Amazon and MySpace, have entered the market with competitive prices and non-DRM music. eMusic is another notable player in the market because of its independent approach to the music download market. It has focused on attracting avid music fans with non-DRM music from independent artists that represent many non-mainstream music genres (e.g., electronica, underground rap, experimental music).³ They currently maintain a corpus of 3.5 million songs and have contributions from 27,000 independent record labels [Nevins (2008)].

Like (illegal) P2P file share services and (legal) music download sites, social networks represent a third, and potentially more significant, Internet development that is changing how music is distributed. Myspace, which was bought by Rupert Murdoch's News Corporation in July of 2005 for \$580 million U.S. dollars, claims that it has 5 million "artist pages" where musicians can publicly share their music [Sandoval (2008)]. In April of 2008, Myspace announced the launch of a music service that will offer

³An *independent* artist is an artist that is not signed by one of the 'big four' music companies: Sony BMG Music Entertainment, Warner Music Group Corp, Universal Music, and EMI.

non-DRM MP3 downloads, ad-supported streaming music, cellphone ringtones, music merchandise, and concert tickets. This service has the backing of three of the four major record labels, including Universal which has a pending lawsuit filed against Myspace for copyright infringement.

Last.fm, which was bought by CBS Interactive for \$280 million U.S. dollars in May of 2007, is a music-specific social network that depends on its users to generate web content: artist biographies and album reviews are created using a public wiki, popularity charts are based on user listening habits, song and artist recommendations are based on collaborative filtering, and *social* tags are collected using a text box in the audio player interface. As of February 2008, Last.fm reported that its database contained information for 150 million distinct tracks by 16 million artists. It provides audio content from big music companies (Universal, EMI, Warner Music Group, Sony BMG, CD Baby), music aggregators (The Orchard, IODA), and over 150,000 independent artists and record labels [Miller et al. (2008)].

Other notable social networks include Imeem, iLike, Mog.com, and the Hype Machine. Imeem was created by original Napster founders and focuses on the sharing of user-generated playlists called “social mix tapes”. iLike, which is funded by Ticketmaster, focuses on social music recommendation and concert promotion. Mog.com was built around music blogs by a large group of music savants. The Hype Machine is a music blog aggregator that continuously crawls and organizes both text and audio content on the web. In addition to these companies, there are hundreds of music-related web-based companies, many of which were launched within the last two years.⁴

1.1.3 Consumption

In the previous section, we described the magnitude of the growing supply of available music, which some experts claim will exceed a billion tracks by tens of mil-

⁴In their 2007 ISMIR *Music Recommendation Tutorial*, Lamere and Celma presented a list of 136 music-related Internet companies. This list does not include websites from record labels or Internet radio stations [Lamere and Celma (2007)].

lions of artists within the next couple years [Lamere and Celma (2007)]. While these numbers seem staggering, the demand for music is equally large. Some illustrative statistics include:

- Between April 2003 and April 2008, Apple sold over 4 billion songs to more than 50 million customers and had a catalog of over 6 million songs [Kaplan (2008)].
- Within the first two weeks of launching its application on Facebook, iLike had registered 3 million new users to their music-oriented social network. By April of 2008, it had over 23 million monthly users [Snider (2008)].
- In February of 2008, Last.fm claimed to have 20 million unique active users from 240 countries per month. They also log 600 million song-play events each month [Miller et al. (2008)].
- In April 2008, MySpace claimed to have 110 million users, 30 million of which actively listen to music on MySpace [Sandoval (2008)].

Demand for music has also been driven by the development of new consumer electronics: personal MP3 players, cellphones, and handheld wireless devices. Apple's line of iPod and iPhone personal MP3 players sold over 140 million units between October 2001 and April 2008. The most recent iPods can store 160 gigabytes of music which is roughly equivalent to two week's worth of continuous MP3 audio content (encoded at 128 kilobytes per second). Other MP3 players include Microsoft's Zune, Creative's Zen, and SanDisk's Sansa. In addition, most new cell phone and handheld wireless devices can play MP3s, and some cell phone providers offer streaming music services. It should also be noted that the cellphone ringtone market in the United States in 2007 was \$550 million dollars [Garrity (2008)].

Consumers are also listening to more music on their personal computers using Internet radio and on-demand music access sites. Pandora revolutionized Internet radio by offering personalized streams of recommended music. A user suggests a known

song or artist and Pandora creates an ad-supported stream of similar music. Music similarity is based on the analysis of human experts who annotate songs using a set of 400 music concepts.⁵ The simplicity of the user interface, as well as the quality of the music recommendations have resulted in a user-base of 11 million individuals. Other major Internet radio companies include Yahoo Launchcast, Slacker, and AccuRadio. Rhapsody, (the rebranded) Napster and a handful of other companies offer subscription-based on-demand access to music. In addition, companies like Seeqpod, Deezer, and YouTube provide free (but potentially illegal) access to music and music videos that are found using webcrawlers or posted by users.

1.2 Music Search and Music Discovery

Given that there are millions of songs by millions of artists, there is a need to develop technologies that help consumers find music. We can identify two distinct use cases: *music search* and *music discovery*. Music search is useful when users know which song, album or artist that they want to find. For example, a friend tells you that the new R.E.M. album is good and you want to purchase that album from a music download site (e.g., Apple iTunes). Music discovery is a less directed pursuit in which a user is not looking for a specific song or artist, but may have some general criteria that they wish to satisfy when looking for music. For example, I may be trying to write my dissertation and want to find non-vocal bluegrass music that is mellow and not too distracting. While search and discovery are often intertwined, search generally involves retrieving music that is known a priori. Discovery involves finding music previously unknown to the listener. There are many existing approaches to music search and music discovery. They include

- Query-by-Metadata - *Search*

We consider *metadata* to be factual information associated with music. This in-

⁵Pandora's set of music concepts is referred to as the *Music Genome* in their marketing literature.

cludes song titles, album titles, artist or band names, composer names, record labels, awards, and popularity information (e.g., record charts, sales information). We also consider metadata to include any relevant biographical (e.g., “raised by grandmother”), socio-cultural (e.g., “influenced by blues tradition at an early age”), economic (e.g., “busked on the streets to make a living”), chronological (e.g., “born in 1945”), and geographical (e.g., “grew up in London”) information. Music metadata is often stored in a structured database and contains relational data (e.g., “played with the Yardbirds”, “influenced by Robert Johnson”). Query-by-metadata involves retrieving music from a database by specifying a (text-based) query. For example, a user can find “all Eric Clapton songs that were recorded prior to 1991.” The most well-known examples of a query-by-metadata systems are commercial music retailers (e.g., Apple iTunes) and Internet search engines (e.g., Google).

- Query-by-performance - *Search*

In recent years, there has been an academic interest in developing music retrieval systems based on human performance: *query-by-humming* [Dannenberg et al. (2003)], *query-by-beatboxing* [Kapur et al. (2004)], and *query-by-tapping* [Eisenberg et al. (2004)], and *query-by-keyboard* [Typke (2007)]. More recently, websites like Midomi and Musipedia have made query-by-performance interfaces available to the general public. However, it can be difficult, especially for an untrained user, to emulate the tempo, pitch, melody, and timbre well enough to make these systems effective [Dannenberg and Hu (2004)].

- Query-by-fingerprint - *Search*

Like query-by-humming, query-by-fingerprint is a technology that involves recording an audio sample and matching it to a database of songs [Cano et al. (2005)]. However, a fingerprint must be a recording of the original audio content rather than a human-generated imitation. Companies like Shazam and Gracenote offer

services where a customer can use a cellphone to record a song that is playing in a natural environment (e.g., in a bar, at a party, on the radio). The recording is matched against a large database of music fingerprints and the name of the identified song is text-messaged back to the customer's cellphone.

- Recommendation-by-popularity - *Discovery*

The two most common way people discover new music is by listening to AM/FM radio and by watching music television (e.g., MTV, VH1, BET) [Enser (2007)]. Whether it is an obscure up-and-coming band with a grassroots fan base or a well-established artist with the backing of a wealthy record company, exposure on the airwaves is critical for success. This success is measured by radio play, sales numbers and critical acclaim, and is reflected by music charts (e.g., Billboard) and awards (e.g., Grammy). Like record stores, music websites use this information to recommend music to customers. However, unlike record stores, music websites have the ability to be more dynamic because they have access to richer and more up-to-date information. For example, Last.fm records the listening habits of each of their 20 million users around the world. As such, they can build custom record charts based on the listening habits of an individual, on the listening habits of an individual's friends, or on the listening habits of all the individuals who belong to a specific demographic (or psychographic) group.

- Browse-by-genre - *Discovery*

A *music genre* is an ontological construct that is used to relate songs or artists, usually based on acoustic or socio-cultural similarity. Examples range from broad genres like 'Rock' and 'World' to more refined genres like 'Neo-bop' and 'Nu Skool Breaks.' A taxonomy of genres is often represented as a directed asymmetric graph (e.g., graph of jazz influences) or a tree (e.g., hierarchy of genres and subgenres). However, genres can be ill-defined and taxonomies are often organized in an inconsistent manner [Pachet and Cazaly (2000); Aucou-

turier and Pachet (2003)]. Despite the shortcomings, they are commonly used by both individuals and music retailers (e.g., Tower Records, Amazon) to organize collections of music. However, as the size of the music collection grows, a taxonomy of genres will become cumbersome in terms of the number of genres and/or the number of songs that are related to each genre.

- Query-by-similarity - *Discovery*

One of the more natural paradigms for finding music is to make use of known songs or artists. While music similarity can be accessed in a number of ways, it is helpful to focus on three types of similarity: *acoustic similarity*, *social similarity*, and *semantic similarity*.

- Acoustic similarity is accessed through the analysis and comparison of multiple audio signals (e.g., “songs that *sound* similar to Jimi Hendrix’s ‘Voodoo Chile’ ”) [Pampalk (2006); Barrington et al. (2007b)].
- Social similarity, also referred to as *collaborative filtering*, finds music based on the preference ratings or purchase sales records from a large group of users (e.g., “people who like Radiohead also like Coldplay”) [Lamere and Celma (2007)]. This is the approach used by Amazon and Last.fm to recommend music to their customers.
- Semantic similarity uses common semantic information (e.g., common genres, instruments, emotional responses, vocal characteristics, etc.) to measure the similarity between songs or artists [Berenzweig et al. (2004)]. It has the added benefit of allowing users to specify which semantic concepts are most important when determining music similarity.

It is important to note that acoustic similarity is generally determined automatically with signal processing and machine learning. Social and semantic similarity requires that these songs be annotated by humans before similarity can be accessed. Pandora’s recommendation engine can be thought of as being half

acoustic and half semantic similarity since human experts are used to annotate each music track with musically objective concepts.⁶

- Query-by-description - *Discovery*

Individuals often use words to describe music. For example, one might say that “Wild Horses” by the Rolling Stones is “a sad folk-rock tune that features somber strumming of an acoustic guitar and a minimalist use of piano and electric slide guitar.” Such descriptions are full of semantic information that can be useful for music retrieval. More specifically, we can annotate music with *tags*, which are short text-based tokens, such as ‘sad’, ‘folk-rock’, and ‘electric slide guitar.’ Music tags can be collected from humans and generated automatically using an autotagging system. See Chapter 2 for a description of our autotagging system and Chapter 4 for a comparison of tag collection approaches. Query-by-description can also include other types of music information such as the number of beats per minute (BPM) or the musical key of a song.

- Heterogeneous Queries - *Search & Discovery*

We can also combine various query paradigms to construct useful new hybrid query paradigms. For example, in this dissertation, we will describe a system that combines metadata, similarity, and description so that a user can find songs that are ‘mellow acoustic Beatles-like music’ or ‘electrified and intense Beatles-like music’.

While we will focus on query-by-description in this dissertation, it is important to note that a complete approach to music search and discovery involves many (or all) of these retrieval paradigms. Currently, Last.fm comes the closest to covering this space by offering query-by-metadata (artist, song, album, record label), browser-by-genre, query-by-social-similarity, and basic query-by-description (i.e., single tag queries only).

⁶Pandora’s set of concepts can be considered *musically objective* since there is a high degree of inter-subject agreement when their musical experts annotate a song.

While Last.fm does not provide a service for query-by-fingerprint, it uses fingerprinting software when collecting data to determine how often each user plays each song.

1.3 Semantic Music Discovery Engine Architecture

In this section, we present the backend architecture for our semantic music discovery engine (see Figure 1.1). The main purpose of the backend is to build a *music information index*. Using this index, music can be retrieved in an efficient manner using a diverse set of descriptive concepts. In Section 1.4, we describe a frontend prototype for the engine to highlight ways in which the music information index is useful for music discovery.

The architecture for the discovery engine can be broken down into three conceptual stages: information *collection*, information *extraction*, and music discovery. First we collect music (i.e., audio tracks) and music information (e.g., metadata, web documents, music tags) using a variety of *data sources* (e.g., websites, surveys, games). These *human annotations* both reflect qualities of the audio content, as well as the social context in which the music is placed. The human annotations and audio content are also used by analytic systems to automatically extract additional information about the music. We then combine both the human annotations and the automatically extracted information to form the *music information index* for each song in our music corpus. This index can then be used for a variety of music discovery tasks: generating music reviews, ranking music by semantic relevance, computing music similarity, building a playlist, clustering artists into groups, etc.

In the following two subsections, we will take a more detailed look at the music information collection and extraction. We will also provide references to related research that specifically pertains to each part of the architecture. In Table 1.1, we outline the structure of the music information index.

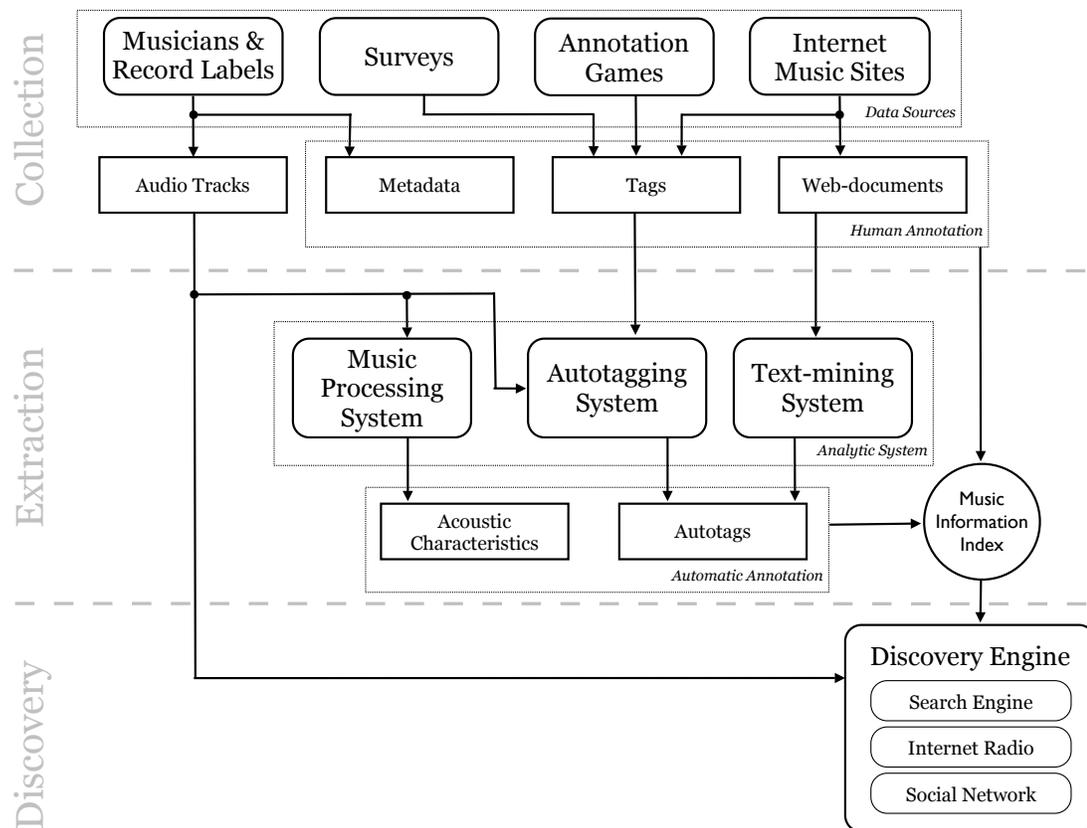


Figure 1.1: Architecture of the Semantic Music Discovery Engine

1.3.1 Information Collection

The most readily-available source of music information is metadata. In general, musician and record labels provide the name of the artist, song, album, and record label. In the context of MP3 files, this information can be encoded directly into the header of the file using the ID3 tags. If the ID3 tags are corrupted or empty, companies like MusicBrainz offer an automatic service where they extract an audio fingerprint from your audio track, match the fingerprint against a large database of audio fingerprints, and send you back the correct metadata. Once the song and artist have been correctly identified, we can collect richer metadata using large relational databases of music information that are maintained by companies like AMG Allmusic and Gracenote. Examples of this metadata include information about the instrumentation, biographical information about the musicians, and popularity information (charts, sales records, and awards).

Collecting music tags, when compared with metadata, is both practically and conceptually more difficult. Practically, given that there are million of songs and thousands of relevant music tags, a major effort to collect even a small percentage of this potential source of semantic information would be required. From a conceptual standpoint, music is inherently subjective in that individuals will often disagree when asked to make qualitative assessments about a song [Berenzweig et al. (2004); McKay and Fujinaga (2006)]. As a result, a tag cannot be thought of as a binary (e.g., on/off) label for a song or artist. Instead, we will consider a tag as a real-valued weight that expresses the strength of association between the semantic concept expressed by the tag and a song. However, it should be noted that a single semantic weight will also be overly restrictive since the strength of association will depend on an individual's socio-cultural background, prior listening experience, and current mood or state of mind. Putting these conceptual issues aside, we can identify three practical techniques for collecting music tags: surveys, annotation games, and social tagging websites. These three approaches are discussed and compared in Chapter 4.

Finally, album, artist and concert reviews, artist biographies, song lyrics, and

other music-related text documents are useful sources of semantic information about music. We can collect many such documents from the Internet using a webcrawler. Once collected, a corpus of music documents can be indexed and used by a text search engine. As we will describe in the following subsection (and in Section ??), we can also generate music tags from this corpus.

1.3.2 Information Extraction

Once we have collected audio tracks, metadata, tags, and text documents, we can extract additional information about music using a combination of audio signal processing, machine learning, and text-mining. We use three types of analytic systems to extract this information:

Music Processing System

For each song, we can calculate a number of specific *acoustic characteristics* by processing the audio track. Some of the more human-usable characteristics include:

- Psychoacoustic - silence, noise, energy, roughness, loudness, and sharpness [McKinney and Breebaart (2003)]
- Rhythmic - tempo (e.g., beats-per-minute BPM), meter (e.g., 4/4 time), rhythmic patterns (e.g., Cha Cha, Viennese Waltz), and rhythmic deviations (e.g., swing factor) [Gouyon and Dixon (2006)]
- Harmonic - key (e.g, A, A#, B, ...), modes (e.g., major/minor), and pitch (e.g., fundamental frequency) [Peeters (2006)]
- Structural - length and segment locations (e.g., chorus detection) [Turnbull et al. (2007b); Goto (2003)]

Each of these characteristics is extracted with a custom digital signal processing algorithm. It should be noted that many such algorithms produce unreliable measurements.

In addition, using some of the characteristics effectively for music retrieval will require a deep level of musical sophistication.

AudioClas is an example of an audio search engine (music samples and sound effects) that allows for searches based on the amount of silence, perceptual roughness, pitch, periodicity, and velocity [Cano (2006)] in an audio file. ‘Smart’ music editors, such as the Sound Palette [Vinet et al. (2002)] and the Sonic Visualizer [Cannam et al. (2006)], also calculate some of these features and use them to annotate audio tracks.

Autotagging System

While the music tags collected from surveys, annotations games, and social tagging sites provide some tags for some songs, the vast majority of songs will be partially or completely unannotated. This is a significant problem since our discovery engine will only be able to retrieve annotated songs. This is referred to as the *cold start* problem and is discussed at length in Chapter 4. In attempt to remedy the cold start problem, we have designed an *autotagging* system that automatically annotates a song with tags based on an analysis of the audio signal. The system is trained using songs that have been manually annotated by humans.

Early work on this topic focused (and continues to focus) on music classification by genre, emotion, and instrumentation (e.g., [Tzanetakis and Cook (2002); Li and Tzanetakis (2003); Essid et al. (2005)]). These classification systems effectively ‘tag’ music with class labels (e.g., “blues,” “sad,” “guitar”). More recently, autotagging systems have been developed to annotate music with a larger, more diverse vocabulary of (non-mutually exclusive) tags [Turnbull et al. (2008); Eck et al. (2007); Sordo et al. (2007)]. In Chapter 2, we present a system that uses a generative approach that learns a Gaussian mixture model (GMM) distribution over an audio feature space for each tag in the vocabulary. Eck et al. use a discriminative approach by learning a boosted decision stump classifier for each tag [Eck et al. (2007)]. Sordo et al. present a non-parametric approach that uses a content-based measure of music similarity to propagate tags from

annotated songs to similar songs that have not been annotated [Sordo et al. (2007)].

Text-mining System

We can also use music-related text documents to automatically generate tags for music. For examples, if many documents (e.g., album reviews, biographies) related to B.B. King have the tag “blues” somewhere in the text, we extract “blues” as a tag for B.B. King’s music. In Section ??, we present a text-mining system that generates tags for music using a large corpus text-document. Our system is based on the research of both Knees [Knees et al. (2008)] and Whitman [Whitman (2005)].

1.3.3 Music Information Index

By putting the human and computer generated annotations together, we create a data structure that can be used for various music discovery tasks. This *music information index* is summarized in Table 1.1.

Table 1.1: Summary of Music Information Index

Acoustic Characteristics	human-usable features are calculated from the audio signal
Documents	indexed set of music-related text-documents
Metadata	factual and relational data about the song or artist
Tags	one tag vector for each annotation approach (e.g., human tags, autotags, text-mined tags)

1.4 CAL Music Discovery Engine

To explore some of the capabilities of the music discovery engine backend, we built a frontend prototype called the *Computer Audition Lab (CAL) Music Discovery Engine*. We provide screenshots of the discovery engine in Figures 1.2-1.4.



Figure 1.2: CAL Music Discovery Engine: Main Page

Using the autotagging system, which will be presented in Chapter 2, we automatically index a corpus of 12,612 songs. Using the web-based interface, a user can specify a text-based query consisting of metadata (album, artist, or song name) and/or music tags (see Figure 1.3 (top)). For example, a user might want to find “Beatles’ songs that are calming, feature an acoustic guitar, and are in the folk tradition.” The system uses the metadata information (e.g., ‘Beatles’) to filter out songs that are not requested by the user. Then, we rank-order the remaining tracks using the music tags (e.g., “calming,” “acoustic guitar,” “folk”). Lastly, we display a list of the most relevant songs (Figure 1.3 (bottom)). Each song is displayed with summary information that is useful for efficient browsing and novel music discovery. The summary includes a playable sample of the audio content, metadata (song, artist, album, release year), an automatically-generated music review that describes the semantic content of the song, and a list of three similar songs. A user can also launch a *semantic radio station* based on their query or any of the songs found on the search results page (see Figure 1.4).

CAL Music Discovery Engine

UCSD **CAL Music Discovery Engine** 

Metadata Search Go Semantic Search Go [Combo Search](#)

Metadata Filter: pick songs by song title, artist name, and album title Go

Song Title contains	<input type="text"/>	does not contain	<input type="text"/>
Artist Name contains	beatles	does not contain	<input type="text"/>
Album Title contains	<input type="text"/>	does not contain	<input type="text"/>
Song, Artist, or Album contains	<input type="text"/>	does not contain	<input type="text"/>

Semantic Ranking: order songs by musical characteristics Go

Musical Genre	Instrumentation	Emotional Content
Alternative <input type="checkbox"/> yes	Acoustic Guitar <input checked="" type="checkbox"/> yes	Aggressive <input type="checkbox"/> yes
Bebop <input type="checkbox"/> yes	Ambient Sounds <input type="checkbox"/> yes	Annoying <input type="checkbox"/> yes
Bluegrass <input type="checkbox"/> yes	Backing vocals <input type="checkbox"/> yes	Arousing <input type="checkbox"/> yes
Blues <input type="checkbox"/> yes	Bass <input type="checkbox"/> yes	Bizarre <input type="checkbox"/> yes
Brit Pop <input type="checkbox"/> yes	Distorted Electric Guitar <input type="checkbox"/> yes	Boring <input type="checkbox"/> yes
Classic Rock <input type="checkbox"/> yes	Electric Guitar <input type="checkbox"/> yes	Calming <input checked="" type="checkbox"/> yes
Cool Jazz <input type="checkbox"/> yes	Female Lead Vocals <input type="checkbox"/> yes	Carefree <input type="checkbox"/> yes
Country <input type="checkbox"/> yes	Hand Drums <input type="checkbox"/> yes	Cheerful <input type="checkbox"/> yes
Dance Pop <input type="checkbox"/> yes	Harmonica <input type="checkbox"/> yes	Emotionless <input type="checkbox"/> yes
Electronica <input type="checkbox"/> yes	Horn Section <input type="checkbox"/> yes	Gloomy <input type="checkbox"/> yes
Folk <input checked="" type="checkbox"/> yes	Male Lead Vocals <input type="checkbox"/> yes	Happy <input type="checkbox"/> yes

CAL Music Discovery Engine

UCSD **CAL Music Discovery Engine** 

Metadata Search Go Semantic Search Go [Combo Search](#)

Combo Search:
Metadata Filtering - 'beatles',
Semantic Ranking - 'Folk', 'Acoustic Guitar', 'Calming',

Songs Found: 77 (Top 10 shown)

▶ **'Julia'** by **The Beatles** on **The Beatles (The White Album) (disc 1)** (1968)
This is a **folk** song that also has a **country** feel. It is **calming** and **tender**. It features **acoustic guitar**, **piano** and **female lead vocals**. The vocals are **emotional** and **high-pitched**. It is a song with **soft beat** and **low energy** that you might like to listen to while **romancing**.

Similar Songs:

▶ **'Ice'** by **Sarah McLachlan** on **Fumbling Towards Ecstasy**

▶ **'Dead of Winter'** by **Eels** on **Electro-Shock Blues**

▶ **'Ribbons Undone'** by **Tori Amos** on **The Beekeeper** (2005)

▶ **'Yesterday'** by **The Beatles** on **Help!**
This is a **singer/songwriter** song that also has a **country** feel. It is **calming** and **boring**. It features **acoustic guitar**, **saxophone** and **female lead vocals**. The vocals are **emotional** and **high-pitched**. It is a song with **low energy** and **soft beat** that you might like to listen to while **romancing**.

Similar Songs:

▶ **'Rose of Aberdeen'** by **Simon & Garfunkel** on **Sounds of Silence**

▶ **'Moonshiner'** by **Uncle Tupelo** on **89/93: An Anthology** (2002)

▶ **'Where Is the Highway Tonight?'** by **Neil Young** on **Lucky Thirteen**

Figure 1.3: CAL Music Discovery Engine: advanced query (top) and results (bottom) for “Beatles folk ‘acoustic guitar’ calming”



Figure 1.4: CAL Semantic Radio Player: displaying playlist for ‘aggressive rap’ query

Like text-based search engines (e.g., Google), the CAL music discovery engine has been designed to be easy-to-use. In addition, the embedded audio player, clickable metadata and tag links, and radio launch buttons make the site highly interactive. Currently, this system only uses metadata and autotags. Future development will involve incorporating human-generated tags from surveys and annotation games, an indexed corpus of musically-relevant text documents, and additional (relational) metadata that can be obtained from commercial music information databases (e.g., Last.fm, AMG Allmusic, Gracenote).

1.5 Summary

In this chapter, we discussed ways in which technology is affecting how music is being produced, distributed, and consumed. The result has been rapid growth in both the quantity of available music and the amount of music that is consumed. This creates a need for powerful new music search and discovery technologies that connect producers

of music (musicians) with consumers of music (fans). Currently, there are a number of query paradigms that are useful for finding music such as query-by-metadata and query-by-similarity. Each paradigm has its own strengths and limitations.

To address some of these limitations, we have presented the architecture for a semantic music discovery engine. This system collects information from existing data sources (record labels, Internet music sites, surveys, annotation games) and automatically extracts information from the audio files and web documents. The result is a music information index that can be used for a variety of music discovery purposes. For example, the CAL Music Discovery Engine is a prototype that explores query-by-description music search, radio playlist generation, and music similarity analysis.

In Chapter 2, we will fully describe and rigorously evaluate our autotagging system. In Chapter 3, we describe Listen Game, which is a web-based multiplayer music annotation game. This game has been developed as a scaleable approach to collecting tags for music. These tags are useful both as indices for our music discovery system and as training data for our autotagging system. In Chapter 4, we compare and contrast alternative approaches for collecting and generating tags for music. In the final chapter, we conclude with a discussion of open research problems and future research directions.

Chapter 2

Using Computer Audition to Generate Tags for Music

2.1 Introduction

Music is a form of communication that can represent human emotions, personal style, geographic origins, spiritual foundations, social conditions, and other aspects of humanity. Listeners naturally use words in an attempt to describe what they hear even though two listeners may use drastically different words when describing the same piece of music. However, words related to some aspects of the audio content, such as instrumentation and genre, may be largely agreed upon by a majority of listeners. This agreement suggests that it is possible to create a computer audition system that can learn the relationship between audio content and words. In this chapter, we describe such a system and show that it can both *annotate* novel audio content with semantically meaningful words and *retrieve* relevant audio tracks from a database of unannotated tracks given a text-based query.

We view the related tasks of semantic annotation and retrieval of audio as one supervised multi-class, multi-label learning problem. We learn a joint probabilistic model of audio content and tags (i.e., short text-based tokens) using an annotated corpus of

audio tracks. Each track is represented as a set of feature vectors that is extracted by passing a short-time window over the audio signal. The text description of a track is represented by an *annotation vector*, a vector of weights where each element indicates how strongly a semantic concept (i.e., a tag) applies to the audio track.

Our probabilistic model is one *tag-level* distribution over the audio feature space for each tag in our vocabulary. Each distribution is modeled using a multivariate Gaussian mixture model (GMM). The parameters of a tag-level GMM are estimated using audio content from a set of training tracks that are positively associated with the tag. Using this model, we can infer likely semantic annotations given a novel track and can use a text-based query to rank-order a set of unannotated tracks. For illustrative purposes, Table 2.1 displays annotations of songs produced by our system. Placing the most likely tags from specific semantic categories into a natural language context demonstrates how our annotation system can be used to generate automatic music reviews. Table 2.7 shows some of the top songs that the system retrieves from our data set, given various text-based queries.

Our model is based on the supervised multi-class labeling (SML) model that has been recently proposed for the task of image annotation and retrieval by Carneiro and Vasconcelos [Carneiro and Vasconcelos (2005)]. They show that their *mixture hierarchies* Expectation Maximization (EM) algorithm [Vasconcelos (2001)], used for estimating the parameters of the tag-level GMMs, is superior to traditional parameter estimation techniques in terms of computational scalability and annotation performance. We confirm these findings for audio data and extend this estimation technique to handle real-valued (rather than binary) class labels. Real-valued class labels are useful in the context of music since the strength of association between a tag and a song is not always all or nothing. For example, based on a study described below, we find that three out of four college students annotate Elvis Presley’s “Heartbreak Hotel” as being a ‘blues’ song while everyone identified B.B. King’s “Sweet Little Angel” as being a blues song. Our *weighted* mixture hierarchies EM algorithm explicitly models these respective strengths of associations when estimating the parameters of a GMM.

Table 2.1: Automatic annotations generated using the audio content. Tags in **bold** are output by our system and then placed into a manually-constructed natural language template.

<p>Frank Sinatra - Fly me to the moon</p> <p>This is a jazzy, singer / songwriter song that is calming and sad. It features acoustic guitar, piano, saxophone, a nice male vocal solo, and emotional, high-pitched vocals. It is a song with a light beat and a slow tempo that you might like listen to while hanging with friends.</p>
<p>Creedence Clearwater Revival - Travelin' Band</p> <p>This is a rockin', classic rock song that is arousing and powerful. It features clean electric guitar, backing vocals, distorted electric guitar, a nice distorted electric guitar solo, and strong, duet vocals. It is a song with a catchy feel and is very danceable that you might like listen to while driving.</p>
<p>New Order - Blue Monday</p> <p>This is a poppy, electronica song that is not emotional and not tender. It features sequencer, drum machine, synthesizer, a nice male vocal solo, and altered with effects, high-pitched vocals. It is a song with a synthesized texture and with positive feelings that you might like listen to while at a party.</p>
<p>Dr. Dre (feat. Snoop Dogg) - Nuthin' but a 'G' thang</p> <p>This is dance poppy, hip-hop song that is arousing and exciting. It features drum machine, backing vocals, male vocal, a nice acoustic guitar solo, and rapping, strong vocals. It is a song that is very danceable and with a heavy beat that you might like listen to while at a party.</p>

The semantic annotations used to train our system come from a user study in which we asked participants to annotate songs using a standard survey. The survey contained questions related to different semantic categories, such as emotional content, genre, instrumentation, and vocal characterizations. The music data used is a set of 500 Western popular songs from 500 unique artists, each of which was reviewed by a minimum of three individuals. Based on the results of this study, we construct a

vocabulary of 174 ‘musically-relevant’ semantic tags. The resulting annotated music corpus, referred to as the *Computer Audition Lab 500 (CAL500)* data set, is publicly-available¹ and may be used as a common test set for future research involving semantic music annotation and retrieval.

Though the focus of this work is on music, our system can be used to model other classes of audio data and is scalable in terms of both vocabulary size and training set size. We demonstrate that our system can successfully annotate and retrieve sound effects using a corpus of 1305 tracks and a vocabulary containing 348 tags.

The following section discusses how this work fits into the field of music information retrieval (MIR) and relates to research on semantic image annotation and retrieval. Sections 2.3 and 2.4 formulate the related problems of semantic audio annotation and retrieval, present the SML model, and describe three parameter estimation techniques including the *weighted* mixture hierarchies algorithm. Section 2.5 describes the collection of human annotations for the CAL500 data set. Section 2.6 describes the sound effects data set. Section 2.7 reports qualitative and quantitative results for annotation and retrieval of music and sound effects. The final section presents a discussion of this research and outlines future directions.

2.2 Related work

A central goal of the music information retrieval community is to create systems that efficiently store and retrieve songs from large databases of musical content [Goto and Hirata (2004); Futrelle and Downie (2002)]. The most common way to store and retrieve music uses metadata such as the name of the composer or artist, the name of the song or the release date of the album. We consider a more general definition of musical metadata as any non-acoustic representation of a song. This includes genre and instrument labels, song reviews, ratings according to bipolar adjectives (e.g., happy/sad), and purchase sales records. These representations can be used as input to collaborative

¹The CAL500 data set can be downloaded from <http://cosmal.ucsd.edu/cal>.

Table 2.2: Music retrieval examples. Each tag (in quotes) represents a text-based query taken from a semantic category (in parenthesis) .

Query	Top 5 Retrieved Songs
'Tender' (Emotion)	Chet Baker - These foolish things Saros - Prelude Norah Jones - Don't know why Art Tatum - Willow weep for me Crosby Stills and Nash - Guinnevere
'Hip Hop' (Genre)	Nelly - Country Grammar C+C Music Factory - Gonna make you sweat Dr. Dre (feat. Snoop Dogg) - Nuthin' but a 'G' thang 2Pac - Trapped Busta Rhymes - Woo hah got you all in check
'Sequencer' (Instrument)	Belief Systems - Skunk werks New Order - Blue Monday Introspekt - TBD Propellerheads - Take California Depeche Mode - World in my eyes
'Exercising' (Usage)	Red Hot Chili Peppers - Give it away Busta Rhymes - Woo hah got you all in check Chic - Le freak Jimi Hendrix - Highway chile Curtis Mayfield - Move on up
'Screaming' (Vocals)	Metallica - One Jackalopes - Rotgut Utopia Banished - By mourning Bomb the Bass - Bug powder dust Nova Express - I'm alive

filtering systems that help users search for music. The drawback of these systems is that they require a novel song to be *manually* annotated before it can be retrieved.

Another retrieval approach, called *query-by-similarity*, takes an audio-based query and measures the similarity between the query and all of the songs in a database [Goto and Hirata (2004)]. A limitation of query-by-similarity is that it requires a user to have a useful audio exemplar in order to specify a query. For cases in which no such exemplar is available, researchers have developed *query-by-humming* [Dannenberg et al. (2003)],

-beatboxing [Kapur et al. (2004)], and *-tapping* [Eisenberg et al. (2004)]. However, it can be hard, especially for an untrained user, to emulate the tempo, pitch, melody, and timbre well enough to make these systems viable [Dannenberg and Hu (2004)]. A natural alternative is to describe music using tags, an interface that is familiar to anyone who has used an Internet search engine. A good deal of research has focused on content-based classification of music by genre [McKinney and Breebaart (2003)], emotion [Li and Tzanetakis (2003)], and instrumentation [Essid et al. (2005)]. These classification systems effectively ‘annotate’ music with class labels (e.g., ‘blues’, ‘sad’, ‘guitar’). The assumption of a predefined taxonomy and the explicit labeling of songs into (mutually exclusive) classes can give rise to a number of problems [Pachet and Cazaly (2000)] due to the fact that music is inherently subjective. A more flexible approach [Berenzweig et al. (2004)] measures the similarity between songs using a semantic ‘anchor space’ where each dimension of the space represents a musical genre.

We propose a content-based *query-by-text* audio retrieval system that learns a relationship between acoustic features and tags from a data set of annotated audio tracks. Our goal is to create a more general system that directly models the relationship between audio content and a vocabulary that is less constrained than existing content-based classification systems. The query-by-text paradigm has been largely influenced by work on the similar task of image annotation. We adapt a supervised multi-class labeling (SML) model [Carneiro et al. (2007)] since it has performed well on the task of image annotation. This approach views semantic annotation as one multi-class problem rather than a set of binary one-vs-all problems. A comparative summary of alternative supervised one-vs-all (e.g., [Forsyth and Fleck (1997)]) and unsupervised (e.g., [Blei and Jordan (2003); Feng et al. (2004)]) models for image annotation is presented in [Carneiro et al. (2007)].

Despite interest within the computer vision community, there has been relatively little work on developing ‘query-by-text’ for audio (and specifically music) data. One exception is the work of Whitman et al. ([Whitman (2005); Whitman and Ellis (2004); Whitman and Rifkin (2002)]). Our approach differs from theirs in a number of ways.

First, they use a set of web-documents associated with an *artist* whereas we use multiple *song* annotations for each song in our corpus. Second, they take a one-vs-all approach and learn a discriminative classifier (a support vector machine or a regularized least-squares classifier) for each tag in the vocabulary. The disadvantage of their approach is that the classifiers output scores (or binary decisions) that are hard to compare with one another. That is, it is hard to identify the *most* relevant tags when annotating a novel song. We propose a generative multi-class model that outputs a semantic multinomial distribution over the vocabulary for each song. As we show in Section 2.3, the parameters of the multinomial distribution provide a natural ranking of tags [Carneiro et al. (2007)]. In addition, semantic multinomials are a compact representation of an audio track which is useful for efficient retrieval.

Other query-by-text audition systems ([Slaney (2002b); Cano and Koppenberger (2004)]) have been developed for annotation and retrieval of sound effects. Slaney’s Semantic Audio Retrieval system ([Slaney (2002b,a)]) creates separate hierarchical models in the acoustic and text space, and then makes links between the two spaces for either retrieval or annotation. Cano and Koppenberger propose a similar approach based on nearest neighbor classification [Cano and Koppenberger (2004)]. The drawback of these non-parametric approaches is that inference requires calculating the similarity between a query and every training example. We propose a parametric approach that requires one model evaluation per semantic concept. In practice, the number of semantic concepts is orders of magnitude smaller than the number of potential training data points, leading to a more scalable solution.

2.3 Semantic audio annotation and retrieval

This section formalizes the related tasks of semantic audio annotation and retrieval as a supervised multi-class, multi-label classification problem where each tag in a vocabulary represents a class and each song is labeled with multiple tags. We learn a *tag-level* (i.e., class-conditional) distribution for each tag in a vocabulary by train-

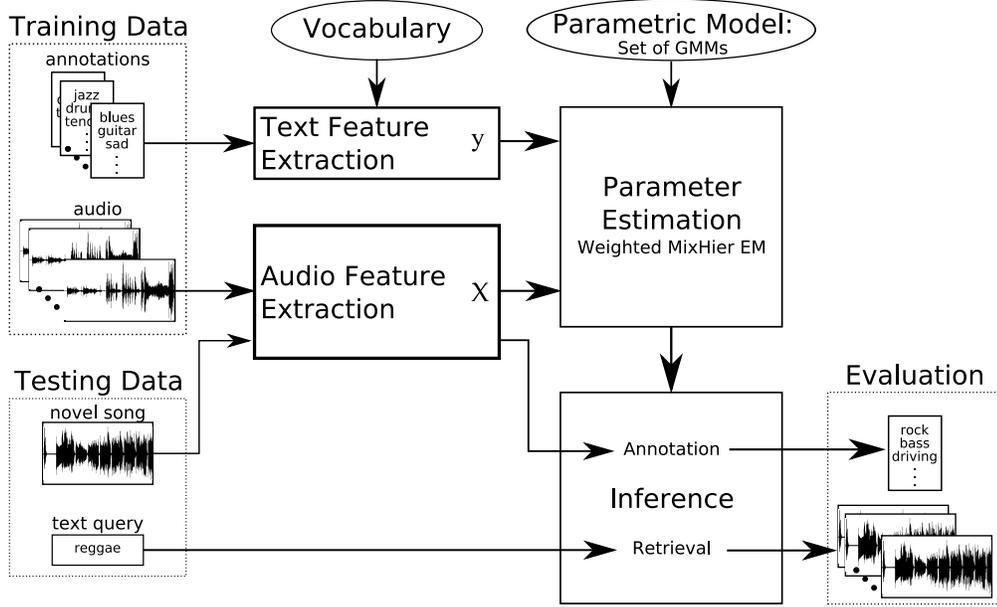


Figure 2.1: Semantic annotation and retrieval model diagram.

ing only on the audio tracks that are positively associated with that tag. A schematic overview of our model is presented in Figure 2.1.

2.3.1 Problem formulation

Consider a vocabulary \mathcal{V} consisting of $|\mathcal{V}|$ unique tags. Each “tag” (or “word”) $w_i \in \mathcal{V}$ is a semantic concept such as “happy”, “blues”, “electric guitar”, “creaky door”, etc. The goal in annotation is to find a set $\mathcal{W} = \{w_1, \dots, w_A\}$ of A semantically meaningful words that describe a query audio track s_q . Retrieval involves rank ordering a set of tracks (e.g., songs) $\mathcal{S} = \{s_1, \dots, s_R\}$ given a set of query words \mathcal{W}_q . It will be convenient to represent the text data describing each song as an *annotation* vector $\mathbf{y} = (y_1, \dots, y_{|\mathcal{V}|})$ where $y_i > 0$ if w_i has a positive semantic association with the audio track and $y_i = 0$ otherwise. The y_i ’s are called *semantic weights* since they are proportional to the strength of the semantic association. If the semantic weights are mapped to $\{0, 1\}$, then they can be interpreted as class labels. We represent an audio track s as a bag $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ of T real-valued feature vectors, where each vector

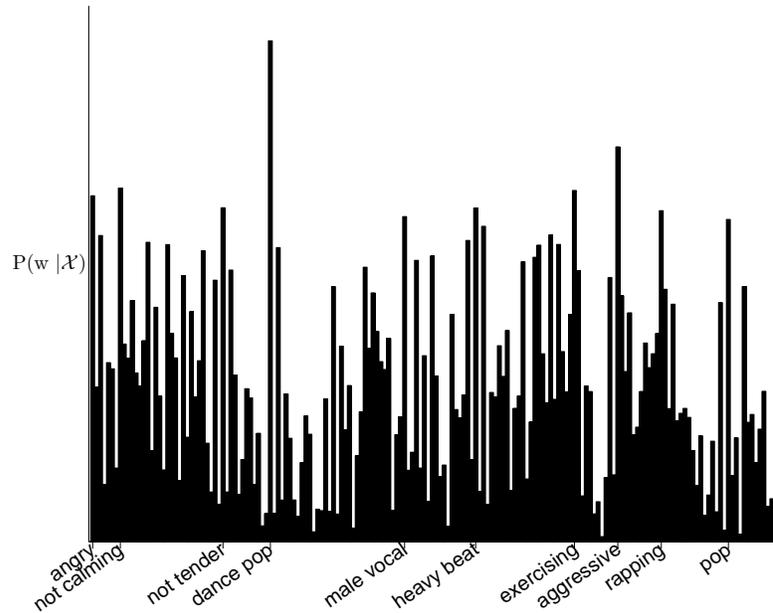


Figure 2.2: Semantic multinomial distribution over all tags in our vocabulary for the Red Hot Chili Pepper’s “Give it Away”; 10 most probable tags are labeled.

\mathbf{x}_t represents features extracted from a short segment of the audio content and T depends on the length of the track. Our data set \mathcal{D} is a collection of track-annotation pairs $\mathcal{D} = \{(\mathcal{X}_1, \mathbf{y}_1), \dots, (\mathcal{X}_{|\mathcal{D}|}, \mathbf{y}_{|\mathcal{D}|})\}$.

2.3.2 Annotation

Annotation can be thought of as a multi-class classification problem in which each tag $w_i \in \mathcal{V}$ represents a class and the goal is to choose the best class(es) for a given audio track. Our approach involves modeling one tag-level distribution over an audio feature space, $P(\mathbf{x}|i)$, for each tag $w_i \in \mathcal{V}$. Given a track represented by the bag-of-feature-vectors $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, we use Bayes’ rule to calculate the posterior probability of each tag in the vocabulary given the audio features:

$$P(i|\mathcal{X}) = \frac{P(\mathcal{X}|i)P(i)}{P(\mathcal{X})}, \quad (2.1)$$

where $P(i)$ is the prior probability that tag w_i will appear in an annotation. We will assume a uniform tag prior, $P(i) = 1/|\mathcal{V}|$ for all $i = 1, \dots, |\mathcal{V}|$, to promote annotation using a diverse set of tags.

To estimate $P(\mathcal{X}|i)$, we assume that \mathbf{x}_a and \mathbf{x}_b are conditionally independent given tag w_i (i.e., $\mathbf{x}_a \perp \mathbf{x}_b | w_i, \forall a, b \leq T, a \neq b$) so that $P(\mathcal{X}|i) = \prod_{t=1}^T P(\mathbf{x}_t|i)$. While this naïve Bayes assumption is unrealistic, attempting to model interactions between feature vectors may be infeasible due to computational complexity and data sparsity. However, ignoring the temporal dependencies tends to underestimate $P(\mathcal{X}|i)$ [Reynolds et al. (2000)]. One common solution is to estimate $P(\mathcal{X}|i)$ with the geometric average $(\prod_{t=1}^T P(\mathbf{x}_t|i))^{\frac{1}{T}}$. This solution has the added benefit of producing comparable probabilities for tracks with different lengths (i.e., when bags-of-feature-vectors do not contain the same number of vectors). That is, longer tracks (with large T) will be, in general, less likely than shorter tracks (with small T) if we use $\prod_{t=1}^T P(\mathbf{x}_t|i)$ to estimate $P(\mathcal{X}|i)$ instead of $(\prod_{t=1}^T P(\mathbf{x}_t|i))^{\frac{1}{T}}$.

We estimate the song prior $P(\mathcal{X})$ by $\sum_{v=1}^{|\mathcal{V}|} P(\mathcal{X}|v)P(v)$ and calculate our final *annotation* equation:

$$P(i|\mathcal{X}) = \frac{(\prod_{t=1}^T P(\mathbf{x}_t|i))^{\frac{1}{T}}}{\sum_{v=1}^{|\mathcal{V}|} (\prod_{t=1}^T P(\mathbf{x}_t|v))^{\frac{1}{T}}}. \quad (2.2)$$

Note that by assuming a uniform tag prior, the $1/|\mathcal{V}|$ factor cancels out of the equation.

Using tag-level distributions ($P(\mathbf{x}|i), \forall i = 1, \dots, |\mathcal{V}|$) and Bayes' rule, we use Equation 2.2 to calculate the parameters of a *semantic multinomial* distribution over the vocabulary. That is, each song in our database is compactly represented as a vector of posterior probabilities $\mathbf{p} = \{p_1, \dots, p_{|\mathcal{V}|}\}$ in a 'semantic space', where $p_i = P(i|\mathcal{X})$ and $\sum_i p_i = 1$. An example of such a semantic multinomial is given in Figure 2.2. To annotate a track with the A best tags, we first calculate the semantic multinomial distribution and then choose the A largest peaks of this distribution, i.e., the A tags with maximum posterior probability.

2.3.3 Retrieval

Given the one-tag query string w_q , a straightforward approach to retrieval involves ranking songs by $P(\mathcal{X}|q)$. However, we find empirically that this approach returns almost the same ranking for every tag in our vocabulary. The problem is due to the fact that many tag-level distributions $P(\mathbf{x}|q)$ are similar (in the Kullback-Leibler sense) to the generic distribution $P(\mathbf{x})$ over the audio feature vector space. This may be caused by using a general purpose audio feature representation that captures additional information besides the specific semantic notion that we are attempting to model. For example, since most of the songs in our training corpus feature vocals, guitar, bass and drums, we would expect most Rolling Stones songs to be more likely than most Louis Armstrong songs with respect to both the generic distribution $P(\mathbf{x})$ and most tag-level distributions $P(\mathbf{x}|q)$. This creates a *track bias* in which generic tracks that have high likelihood under this generic distribution will also have high likelihood under many of the tag-level distributions. Track bias is solved by dividing $P(\mathcal{X}|q)$ by the track prior $P(\mathcal{X})$ to normalize for track bias. Note that, if we assume a uniform tag prior (which doesn't affect the relative ranking), this is equivalent to ranking by $P(q|\mathcal{X})$ which is calculated in Equation 2.2 during annotation. To summarize, we first annotate our audio corpus by estimating the parameters of a semantic multinomial for each track. For a one-tag query w_q , we rank the tracks by the q^{th} parameter of each track's semantic multinomial distribution.

We can naturally extend this approach to multi-tag queries by constructing a *query multinomial* distribution from the tags in the query string. That is, when a user enters a query, we construct a query multinomial' distribution, parameterized by the vector $\mathbf{q} = \{q_1, \dots, q_{|\mathcal{V}|}\}$, by assigning $q_i = C$ if tag w_i is in the text-based query, and $q_i = \epsilon$ where $1 \gg \epsilon > 0$ otherwise. We then normalize \mathbf{q} , making its elements sum to unity so that it correctly parameterizes a multinomial distribution. In practice, we set the $C = 1$ and $\epsilon = 10^{-6}$. However, we should stress C need not be a constant, rather it could be a function of the query string. For example, we may want to give more weight

to tags that appear earlier in the query string as is commonly done by Internet search engines for retrieving web documents. Examples of a semantic query multinomial and the retrieved song multinomials are given in Figure 2.3.

Once we have a query multinomial, we rank all the songs in our database by the Kullback-Leibler (KL) divergence between the query multinomial \mathbf{q} and each semantic multinomial. The KL divergence between \mathbf{q} and a semantic multinomial \mathbf{p} is given by [Cover and Thomas (1991)]:

$$KL(\mathbf{q}||\mathbf{p}) = \sum_{i=1}^{|\mathcal{V}|} q_i \log \frac{q_i}{p_i}, \quad (2.3)$$

where the query distribution serves as the ‘true’ distribution. Since $q_i = \epsilon$ is effectively zero for all tags that do not appear in the query string, a one-tag query w_i reduces to ranking by the i -th parameter of the semantic multinomials. For a multiple-tag query, we only need to calculate one term in Equation 2.3 per tag in the query. This leads to a very efficient and scalable approach for music retrieval in which the majority of the computation involves sorting the D scalar KL divergences between the query multinomial and each song in the database.

2.4 Parameter Estimation

For each tag $w_i \in \mathcal{V}$, we learn the parameters of the tag-level (i.e., class-conditional) distribution, $P(\mathbf{x}|i)$, using the audio features from all tracks that have a positive association with tag w_i . Each distribution is modeled with a R -component mixture of Gaussians distribution parameterized by $\{\pi_r, \mu_r, \Sigma_r\}$ for $r = 1, \dots, R$. The tag-level distribution for tag w_i is given by:

$$P(\mathbf{x}|i) = \sum_{r=1}^R \pi_r \mathcal{N}(\mathbf{x}|\mu_r, \Sigma_r),$$

where $\mathcal{N}(\cdot|\mu, \Sigma)$ is a multivariate Gaussian distribution with mean μ , covariance matrix Σ , and mixing weight π_r . In this work, we consider only diagonal covariance matrices since using full covariance matrices can cause models to overfit the training data

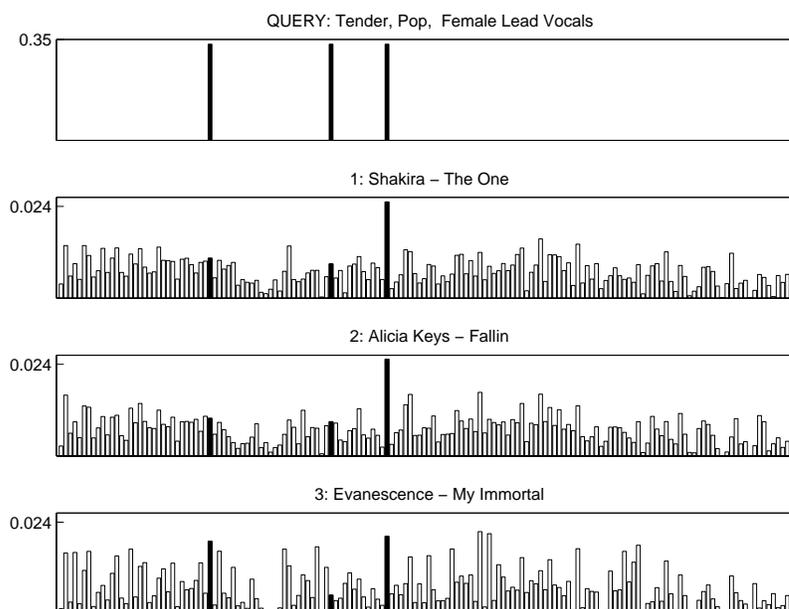


Figure 2.3: Multinomial distributions over the vocabulary of musically-relevant tags. The top distribution represents the *query* multinomial for the three-tag query presented in Table 2.7. The next three distribution are the *semantic* multinomials for top three retrieved songs.

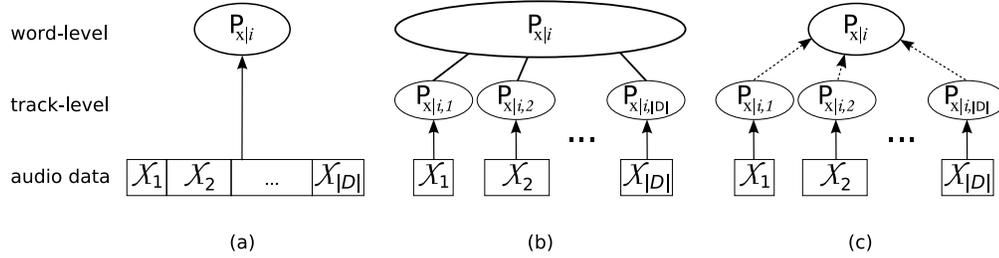


Figure 2.4: (a) Direct, (b) naive averaging, and (c) mixture hierarchies parameter estimation. Solid arrows indicate that the distribution parameters are learned using standard EM. Dashed arrows indicate that the distribution is learned using mixture hierarchies EM. Solid lines indicate weighted averaging of track-level models.

while scalar covariances do not provide adequate generalization. The resulting set of $|\mathcal{V}|$ models each have $\mathcal{O}(R \cdot D)$ parameters, where D is the dimension of feature vector \mathbf{x} .

We consider three parameter estimation techniques for learning the parameters of a tag-level distributions: direct estimation, (weighted) model averaging, and (weighted) mixture hierarchies estimation. The techniques are similar in that, for each tag-level distribution, they use the Expectation-Maximization (EM) algorithm for fitting a mixture of Gaussians to training data. They differ in how they break down the problem of parameter estimation into subproblems and then merge these results to produce a final density estimate.

2.4.1 Direct Estimation

Direct estimation trains a model for each tag w_i using the superset of feature vectors for all the songs that have tag w_i in the associated human annotation: $\bigcup \mathcal{X}_d, \forall d$ such that $[y_d]_i > 0$. Using this training set, we directly learn the tag-level mixture of Gaussians distribution using the EM algorithm (see Figure 2.4a). The drawback of using this method is that computational complexity increases with training set size. We find that, in practice, we are unable to estimate parameters using this method in a reasonable

amount of time since there are on the order of 100,000’s of training vectors for each tag-level distribution. One suboptimal work around to this problem is to simply ignore (i.e., subsample) part of the training data.

2.4.2 Model Averaging

Instead of directly estimating a tag-level distribution for w_i , we can first learn *track-level* distributions, $P(\mathbf{x}|i, d)$ for all tracks d such that $[\mathbf{y}_d]_i > 0$. Here we use EM to train a track-level distribution from the feature vectors extracted from a single track. We then create a tag-level distribution by calculating a weighted average of all the track-level distributions where the weights are set by how strongly each tag w_i relates to that track:

$$P_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|i) = \frac{1}{C} \sum_{d=1}^{|\mathcal{D}|} [\mathbf{y}_d]_i \sum_{k=1}^K \pi_k^{(d)} \mathcal{N}(\mathbf{x}|\mu_k^{(d)}, \Sigma_k^{(d)}),$$

where $C = \sum_d [\mathbf{y}_d]_i$ is the sum of the semantic weights associated with tag w_i , $|\mathcal{D}|$ is total number of training examples, and K is the number of mixture components in each track-level distribution (see Figure 2.4b).

Training a model for each track in the training set and averaging them is relatively efficient. The drawback of this non-parametric estimation technique is that the number of mixture components in the tag-level distribution grows with the size of the training database since there will be K components for each track-level distribution associated with tag w_i . In practice, we may have to evaluate thousands of multivariate Gaussian distributions for each of the feature vectors $\mathbf{x}_t \in \mathcal{X}_q$ of a novel query track, \mathcal{X}_q . Note that \mathcal{X}_q may contain thousands of feature vectors depending on the audio representation.

2.4.3 Mixture Hierarchies

The benefit of direct estimation is that it produces a distribution with a fixed number of parameters. However, in practice, parameter estimation is infeasible without

subsampling the training data. Model averaging efficiently produces a distribution but it is computationally expensive to evaluate this distribution since the number of parameters increases with the size of the training data set. Mixture hierarchies estimation is an alternative that efficiently produces a tag-level distribution with a fixed number of parameters [Vasconcelos (2001)].

Consider the set of $|\mathcal{D}|$ track-level distributions (each with K mixture components) that are learned during model averaging estimation for tag w_i . We can estimate a tag-level distribution with R components by combining the $|\mathcal{D}| \cdot K$ track-level components using the mixture hierarchies EM algorithm (see Figure 2.4c). This EM algorithm iterates between the E-step and the M-step as follows:

E-step: Compute the responsibilities of each tag-level component r to a track-level component k from track d :

$$h_{(d),k}^r = \frac{[\mathbf{y}_d]_i \left[\mathcal{N}(\mu_k^{(d)} | \mu_r, \Sigma_r) e^{-\frac{1}{2} \text{Tr}\{(\Sigma_r)^{-1} \Sigma_k^{(d)}\}} \right]^{\pi_k^{(d)} N} \pi_r}{\sum_l \left[\mathcal{N}(\mu_k^{(d)} | \mu_l, \Sigma_l) e^{-\frac{1}{2} \text{Tr}\{(\Sigma_l)^{-1} \Sigma_k^{(d)}\}} \right]^{\pi_k^{(d)} N} \pi_l},$$

where N is a user defined parameter. In practice, we set $N = K$ so that on average $\pi_k^{(d)} N$ is equal to 1.

M-step: Update the parameters of the tag-level distribution

$$\begin{aligned} \pi_r^{new} &= \frac{\sum_{(d),k} h_{(d),k}^r}{W \cdot K}, \text{ where } W = \sum_{d=1}^{|\mathcal{D}|} [\mathbf{y}_d]_i \\ \mu_r^{new} &= \sum_{(d),k} z_{(d),k}^r \mu_k^{(d)}, \text{ where } z_{(d),k}^r = \frac{h_{(d),k}^r \pi_k^{(d)}}{\sum_{(d),k} h_{(d),k}^r \pi_k^{(d)}}, \\ \Sigma_r^{new} &= \sum_{(d),k} z_{(d),k}^r \left[\Sigma_k^{(d)} + (\mu_k^{(d)} - \mu_r)(\mu_k^{(d)} - \mu_r)^T \right]. \end{aligned}$$

From a generative perspective, a track-level distribution is generated by sampling *mixture components* from the tag-level distribution. The observed audio features are then samples from the track-level distribution. Note that the number of parameters for the tag-level distribution is the same as the number of parameters resulting from direct

estimation yet we learn this model using all of the training data without subsampling. We have essentially replaced one computationally expensive (and often impossible) run of the standard EM algorithm with $|\mathcal{D}|$ computationally inexpensive runs and one run of the mixture hierarchies EM. In practice, mixture hierarchies EM requires about the same computation time as one run of standard EM.

Our formulation differs from that derived in [Vasconcelos (2001)] in that the responsibility, $h_{(d),k}^r$, includes multiplication by the semantic weight $[y_d]_i$ between tag w_i and audio track s_d . This *weighted mixture hierarchies algorithm* reduces to the standard formulation when the semantic weights are either 0 or 1. The semantic weights can be interpreted as a relative measure of importance of each training data point. That is, if one data point has a weight of 2 and all others have a weight of 1, it is as though the first data point actually appeared twice in the training set.

2.5 Semantically Labeled Music Data

Perhaps the fastest and most cost effective way to collect semantic information about music is to mine web documents that relate to songs, albums or artists [Whitman and Rifkin (2002); Turnbull et al. (2006)]. Whitman et al. collect a large number web-pages related to the artist when attempting to annotate individual songs [Whitman and Rifkin (2002)]. One drawback of this methodology is that it produces the same training annotation vector for all songs by a single artist. This is a problem for many artists, such as Paul Simon and Madonna, who have produced an acoustically diverse set of songs over the course of their careers. In previous work, we take a more song-specific approach by text-mining song reviews written by expert music critics [Turnbull et al. (2006)]. The drawback of this technique is that critics do not explicitly make decisions about the relevance of each individual tag when writing about songs and/or artists. In both works, it is evident that the semantic labels are a noisy version of an already problematic ‘subjective ground truth.’

To address the shortcomings of noisy semantic data mined from text-documents,

we decided to collect a ‘clean’ set of semantic labels by asking human listeners to explicitly label songs with acoustically-relevant tags. We considered 135 musically-relevant concepts spanning six semantic categories: 29 instruments were annotated as present in the song or not; 22 vocal characteristics were annotated as relevant to the singer or not; 36 genres, a subset of the Codaich genre list [McKay et al. (2006)], were annotated as relevant to the song or not; 18 emotions, found by Skowronek et al. [Skowronek et al. (2006)] to be both important and easy to identify, were rated on a scale from one to three (e.g., ”not happy”, ”neutral”, ”happy”); 15 song concepts describing the acoustic qualities of the song, artist and recording (e.g., tempo, energy, sound quality); and 15 usage terms from [Hu et al. (2006)] (e.g., “I would listen to this song while *driving, sleeping, etc.*”).

The music corpus is a selection of 500 Western popular songs from the last 50 years by 500 different artists. This set was chosen to maximize the acoustic variation of the music while still representing some familiar genres and popular artists. The corpus includes 88 songs from the Magnatunes database [Buckman (2006)], one from each artist whose songs are not from the classical genre.

To generate new semantic labels, we paid 66 undergraduate students to annotate our music corpus with the semantic concepts from our vocabulary. Participants were rewarded \$10 per hour to listen to and annotate music in a university computer laboratory. The computer-based annotation interface contained a MP3 player and an HTML form. The form consisted of one or more radio boxes and/or check boxes for each of our 135 concepts. The form was not presented during the first 30 seconds of song playback to encourage undistracted listening. Subjects could advance and rewind the music and the song would repeat until they completed the annotation form. Each annotation took about 5 minutes and most participants reported that the listening and annotation experience was enjoyable. We collected at least 3 semantic annotations for each of the 500 songs in our music corpus and a total of 1708 annotations. This annotated music corpus is referred to as the Computer Audition Lab 500 (CAL500) data set.

2.5.1 Semantic Feature Representation

We expand the set of *concepts* to a set of 237 *tags* by mapping all bipolar concepts to two individual tags. For example, ‘tender’ gets mapped to ‘tender’ and ‘not tender’ so that we can explicitly learn separate models for tender songs and songs that are not tender. Note that, according to the data that we collected, many songs may be annotated as neither tender nor not tender. Other concepts, such as genres or instruments, are mapped directly to a single tag.

For each song, we have a collection of human annotations where each annotation is a vector of numbers expressing the response of a subject to a set of tags. For each tag, the annotator has supplied a response of +1 or -1 if the annotator believes the song is or is not indicative of the tag, or 0 if unsure. We take all the annotations for each song and compact them to a single annotation vector by observing the level of agreement over all annotators. Our final semantic weights \mathbf{y} are

$$[\mathbf{y}]_i = \max \left(0, \left[\frac{\#(\text{Positive Votes}) - \#(\text{Negatives Votes})}{\#(\text{Annotations})} \right]_i \right).$$

For example, for a given song, if four annotators have labeled a concept w_i with +1, +1, 0, -1, then $[\mathbf{y}]_i = 1/4$. The semantic weights are used for parameter estimation.

For evaluation purposes, we also create a binary ‘ground truth’ annotation vector for each song. To generate this vector, we label a song with a tag if a minimum of two people vote for the tag and there is a high level of agreement ($[\mathbf{y}]_i \geq .80$) between all subjects. This assures that each positive label is reliable. Finally, we prune all tags that are represented by fewer than five songs. This reduces our set of 237 tags to a set of 174 tags.

2.5.2 Music Feature Representation

Each song is represented as a *bag-of-feature-vectors*: a set of feature vectors where each vector is calculated by analyzing a short-time segment of the audio signal. In particular, we represent the audio with a time series of *Delta-MFCC* feature vectors

[Buchanan (2005)]. A time series of Mel-frequency cepstral coefficient (MFCC) [Rabiner and Juang (1993)] vectors is extracted by sliding a half-overlapping, short-time window (~ 23 msec) over the song’s digital audio file. A Delta-MFCC vector is calculated by appending the first and second instantaneous derivatives of each MFCC to the vector of MFCCs. We use the first 13 MFCCs resulting in about 5,200 39-dimensional feature vectors per minute of audio content. The reader should note that the SML model (a set of GMMs) ignores the temporal dependencies between adjacent feature vectors within the time series. We find that randomly sub-sampling the set of delta cepstrum feature vectors so that each song is represented by 10,000 feature vectors reduces the computation time for parameter estimation and inference without sacrificing overall performance.

We have also explored a number of alternative feature representations, many of which have shown good performance on the task of genre classification, artist identification, song similarity, and/or cover song identification [Downie (2005)]. These include auditory filterbank temporal envelope [McKinney and Breebaart (2003)], dynamic MFCC McKinney and Breebaart (2003), MFCC (without derivatives), chroma features [Ellis and Poliner (2007)], and fluctuation patterns [Pampalk (2006)]. While a detailed comparison is beyond the scope of this paper, one difference between these representations is the amount of the audio content that is summarized by each feature vector. For example, a Delta-MFCC vector is computed from less than 80 msec of audio content, a dynamic MFCC vector summarizes MFCCs extracted over $3/4$ of a second, and fluctuation patterns can represent information extracted from 6 seconds of audio content. We found that Delta-MFCC features outperformed the other representations with respect to both annotation and retrieval performance.

2.6 Semantically Labeled Sound Effects Data

To confirm the general applicability of the SML model to other classes of audio data, we show that we can also annotate and retrieve sound effects. We use the BBC

sound effects library which consists of 1305 sound effects tracks [Slaney (2002b)]. Each track has been annotated with a short 5-10 tag caption. We automatically extract a vocabulary consisting of 348 tags by including each tag that occurs in 5 or more captions. Each caption for a track is represented as a 348-dimensional binary annotation vector where the i -th value is 1 if tag w_i is present in the caption, and 0 otherwise. As with music, the audio content of the sound effect track is represented as a time series of Delta-MFCC vectors, though we use a shorter short-time window (~ 11.5 msec) when extracting MFCC vectors. The shorter time window is used in an attempt to better represent important inharmonic noises that are generally present in sound effects.

2.7 Model evaluation

In this section, we quantitatively evaluate our SML model for audio annotation and retrieval. We find it hard to compare our results to previous work [Slaney (2002b); Cano and Koppenberger (2004); Whitman and Ellis (2004)] since existing results are mainly qualitative and relate to individual tracks, or focus on a small subset of sound effects (e.g., isolated musical instruments or animal vocalizations).

For comparison, we evaluate our two SML models and compare them against three baseline models. The parameters for one SML model, denoted ‘MixHier’, are estimated using the weighted mixture hierarchies EM algorithm. The second SML model, denoted ‘ModelAvg’, results from weighted modeling averaging. Our three baseline models include a ‘Random’ lower bound, an empirical upper bound (denoted ‘UpperBnd’), and a third ‘Human’ model that serves as a reference point for how well an individual human would perform on the annotation task.

The ‘Random’ model samples tags (without replacement) from a multinomial distribution parameterized by the tag prior distribution, $P(i)$ for $i = 1, \dots, |\mathcal{V}|$, estimated using the observed tag counts of a training set. Intuitively, this prior stochastically generates annotations from a pool of the most frequently used tags in the training set. The ‘UpperBnd’ model uses the ground truth to annotated songs. However, since we require

that each model use a fixed number of tags to annotate each song, if the ground truth annotation contains too many tags, we randomly pick a subset of the tags from the annotation. Similarly, if the ground truth annotation contains too few tags, we randomly add tags to the annotation from the rest of the vocabulary.

Lastly, we will compare an individual’s annotation against a ‘ground truth’ annotation that is found by averaging multiple annotations (i.e., an annotation based on group consensus). Specifically, the ‘Human’ model is created by randomly holding out a single annotation for a song that has been annotated by 4 or more individuals. This model is evaluated against a ‘ground truth’ that is obtained combining the remaining annotations for that song. (See Section 2.5.1 for the details of our summarization process.) It should be noted that each individual annotation uses on average 36 of the 174 tags in our vocabulary. Each ground truth annotation uses on average only 25 tags since we require a high-level of agreement between multiple independent annotators for a tag to be considered relevant. This reflects the fact that music is inherently subjective in that individuals use different tags to describe the same song.

2.7.1 Annotation

Using Equation 2.2, we annotate all test set songs with 10 tags and all test set sound effect tracks with 6 tags. Annotation performance is measured using mean *per-tag* precision and recall. Per-tag precision is the probability that the model correctly uses the tag when annotating a song. Per-tag recall is the probability that the model annotates a song that should have been annotated with the tag. More formally, for each tag w , $|w_H|$ is the number of tracks that have tag w in the human-generated ‘ground truth’ annotation. $|w_A|$ is the number of tracks that our model automatically annotates with tag w . $|w_C|$ is the number of ‘correct’ tags that have been used both in the ground truth annotation and by the model. Per-tag recall is $|w_C|/|w_H|$ and per-tag precision is $|w_C|/|w_A|^2$.

²If the model never annotates a song with tag w then per-tag precision is undefined. In this case, we estimate per-tag precision using the empirical prior probability of the tag $P(i)$. Using the prior is similar to using the ‘Random’ model to estimate the per-tag precision, and thus, will in general hurt model performance. This produces a desired effect since we are interested in designing a model that annotates

While trivial models can easily maximize one of these measures (e.g., labeling all songs with a certain tag or, instead, none of them), achieving excellent precision and recall simultaneously requires a truly valid model.

Mean per-tag recall and precision is the average of these ratios over all the tags in our vocabulary. It should be noted that these metrics range between 0.0 and 1.0, but one may be upper-bounded by a value less than 1.0 if either the number of tags that appear in a ground truth annotation is greater or lesser than the number of tags that are output by our model. For example, if our system outputs 10 tags to annotate a test song where the ground truth annotation contains 25 tags, mean per-tag recall will be upper-bounded by a value less than one. The exact upper bounds for recall and precision depend on the relative frequencies of each tag in the vocabulary and can be empirically estimated using the ‘UpperBnd’ model which is described above.

It may seem more straightforward to use *per-song* precision and recall, rather than the per-tag metrics. However, per-song metrics can lead to artificially good results if a system is good at predicting the few common tags relevant to a large group of songs (e.g., ‘rock’) and bad at predicting the many rare tags in the vocabulary. Our goal is to find a system that is good at predicting all the tags in our vocabulary. In practice, using the 10 best tags to annotate each of the 500 songs, our system outputs 166 of the 174 tags for at least one song.

Table 2.3 presents quantitative results for music and Table 2.4 for sound effects. Table 2.3 also displays annotation results using only tags from each of six semantic categories (emotion, genre, instrumentation, solo, usage and vocal). All reported results are means and standard errors computed from 10-fold cross-validation (i.e., 450-song training set, 50-song test set).

The quantitative results demonstrate that the SML models trained using model averaging (ModelAvg) and mixture hierarchies estimation (MixHier) significantly outperform the random baselines for both music and sound effects. For music, MixHier significantly outperforms ModelAvg in both precision and recall when considering the songs using many tags from our vocabulary.

entire vocabulary as well as showing superior performance for most semantic categories, where ‘instrumentation precision’ is the sole exception. However, for sound effects, ModelAvg significantly outperforms MixHier. This might be explained by interpreting model averaging as a non-parametric approach in which the likelihood of the query track is computed under every track-level model in the database. For our sound effects data set, it is often the case that semantically related pairs of tracks are acoustically very similar causing that one track-level model to dominate the average.

Over the entire music vocabulary, the MixHier model performance is comparable to the Human model. It is also interesting to note that MixHier model performance is significantly worse than the Human model performance for the more ‘objective’ semantic categories (e.g., Instrumentation and Genre) but is comparable for more ‘subjective’ semantic categories (e.g., Usage and Emotion). We are surprised by the low Human model precision, especially for some of these more objective categories, when compared against the UpperBnd model. Taking a closer look at precision for individual tags, while there are some tags with relatively high precision, such as ‘male lead vocals’ (0.96) and ‘drum set’ (0.81), there are many tags with low precision. Low precision tags arise from a number of causes including test subject inattentiveness (due to boredom or fatigue), non-expert test-subjects (e.g., can’t detect a ‘trombone’ in a horn section), instrument ambiguity (e.g., deciding between ‘acoustic guitar’ vs. ‘clean electric guitar’), and our summarization process. For example, consider the tag ‘clean electric guitar’ and the song “Everything she does is magic” by The Police. Given four test subjects, two subjects positively associate the song with the tag because the overall guitar sound is clean, one is unsure, and one says there is no ‘clean electric guitar’ presumably because, technically, the guitarist makes use of a delay distortion³. Our summarization process would not use the tag to label this songs despite the fact that half of the subjects used this tag to describe the song. In Section 2.8, we will discuss both ways to improve the survey process as well as an alternative data collection technique.

³A delay causes the sound to repeatedly echo as the sound fades away, but does not grossly distort the timbre of electric guitar.

2.7.2 Retrieval

For each one-tag query w_q in \mathcal{V} , we rank-order a test set of songs. For each ranking, we calculate the average precision (AP) [Feng et al. (2004)] and the area under the receiver operating characteristic curve (AROC). Average precision is found by moving down our ranked list of test songs and averaging the precisions at every point where we correctly identify a new song. An ROC curve is a plot of the true positive rate as a function of the false positive rate as we move down this ranked list of songs. The area under the ROC curve (AROC) is found by integrating the ROC curve and is upper-bounded by 1.0. Random guessing in a retrieval task results in an AROC of 0.5. Comparison to human performance is not possible for retrieval since an individual’s annotations do not provide a ranking over all retrievable audio tracks. Mean AP and Mean AROC are found by averaging each metric over all the tags in our vocabulary (shown Tables 2.8 and 2.6).

As with the annotation results, we see that our SML models significantly outperform the random baseline and that MixHier outperforms ModelAvg for music retrieval. For sound effects retrieval, MixHier and ModelAvg are comparable if we consider Mean AROC, but MixHier shows superior performance if we consider Mean AP.

2.7.3 Multi-tag Retrieval

We evaluate every one-, two-, and three-tag query drawn from a subset of 159 tags from our 174-tag vocabulary. (The 159 tags are those that are used to annotate 8 or more songs in our 500-song corpus.) First, we create query multinomials for each query string as described in Section 2.3.3. For each query multinomial, we rank order the 500 songs by the KL divergence between the query multinomial and the semantic multinomials generated during annotation. (As described in the previous subsection, the semantic multinomials are generated from a test set using cross-validation and can be considered representative of a novel test song.)

Table 2.7 shows the top 5 songs retrieved for a number of text-based queries. In

addition to being (mostly) accurate, the reader should note that queries, such as ‘Tender’ and ‘Female Vocals’, return songs that span different genres and are composed using different instruments. As more tags are added to the query string, note that the songs returned are representative of all the semantic concepts in each of the queries.

By considering the “ground truth” target for a multiple-tag query as all the songs that are associated with *all* the tags in the query string, we can quantitatively evaluate retrieval performance. Columns 3 and 4 of Table 2.8 show MeanAP and MeanAROC found by averaging each metric over all testable one, two and three tag queries. Column 1 of Table 2.8 indicates the proportion of all possible multiple-tag queries that actually have 8 or more songs in the ground truth against which we test our model’s performance.

As with the annotation results, we see that our model significantly outperforms the random baseline. As expected, MeanAP decreases for multiple-tag queries due to the increasingly sparse ground truth annotations (since there are fewer relevant songs per query). However, an interesting finding is that the MeanAROC actually increases with additional query terms, indicating that our model can successfully integrate information from multiple tags.

2.7.4 Comments

The qualitative annotation and retrieval results in Tables 2.7 and 2.1 indicate that our system produces sensible semantic annotations of a song and retrieves relevant songs, given a text-based query. Using the explicitly annotated music data set described in Section 2.5, we demonstrate a significant improvement in performance over similar models trained using weakly-labeled text data mined from the web [Turnbull et al. (2006)] (e.g., music retrieval MeanAROC increases from 0.61 to 0.71). The CAL500 data set, automatic annotations of all songs, and retrieval results for each tag, can be found at the UCSD Computer Audition Lab website (<http://cosmal.ucsd.edu/cal>).

Our results are comparable to state-of-the-art content-based image annotation systems [Carneiro et al. (2007)] which report mean per-tag recall and precision scores

of about 0.25. However, the relative objectivity of the tasks in the two domains as well as the vocabulary, the quality of annotations, the features, and the amount of data differ greatly between our audio annotation system and existing image annotation systems making any direct comparison dubious at best.

2.8 Discussion and Future Work

The qualitative annotation and retrieval results in Tables 2.1 and 2.7 indicate that our system can produce sensible semantic annotations for an acoustically diverse set of songs and can retrieve relevant songs given a text-based query. When comparing these results with previous results based on models trained using web-mined data [Turnbull et al. (2006)], it is clear that using ‘clean’ data (i.e., the CAL500 data set) results in much more intuitive music reviews and search results.

Our goal in collecting the CAL500 data set was to quickly and cheaply collect a small music corpus with reasonably accurate annotations for the purposes of training our SML model. The human experiments were conducted using (mostly) non-expert college students who spent about five minutes annotating each song using our survey. While we think that the CAL500 data set will be useful for future content-based music annotation and retrieval research, it is not of the same quality as data that might be collected using a highly-controlled psychoacoustics experiment. Future improvements would include spending more time training our test subjects and inserting consistency checks so that we could remove inaccurate annotations from test subjects who show poor performance.

Currently, we are looking at two extensions to our data collection process. The first involves vocabulary selection: if a tag in the vocabulary is inconsistently used by human annotators, or the tag is not clearly represented by the underlying acoustic representation, the tag can be considered as *noisy* and should be removed from the vocabulary to denoise the modeling process. We explore these issues in [Torres et al. (2007)], whereby we devise vocabulary pruning techniques based on measurements of human agreement and correlation of tags with the underlying audio content.

Our second extension involves collecting a much larger annotated data set of music using web-based human computation games [Turnbull et al. (2007c)]. We have developed a web-based game called “Listen Game” which allows multiple ‘annotators’ to label music through realtime competition. We consider this to be a more scalable and cost-effective approach for collecting high-quality music annotations than laborious surveys. We are also able to grow our vocabulary by allowing users to suggest tags that describe the music.

When compared with direct estimation and model averaging, our weighted mixture hierarchies EM is more computationally efficient and produces density estimates that result in better end performance. The improvement in performance may be attributed to the fact that we represent each track with a track-level distribution before modeling a tag-level distribution. The track-level distribution is a smoothed representation of the bag-of-feature-vectors that are extracted from the audio signal. We then learn a mixture from the mixture components of the track-level distributions that are semantically associated with a tag. The benefit of using smoothed estimates of the tracks is that the EM framework, which is prone to find poor local maxima, is more likely to converge to a better density estimate.

The *semantic multinomial* representation of a song, which is generated during annotation (see Section 2.3.2), is a useful and compact representation of a song. In derivative work [Turnbull et al. (2007a)], we show that if we construct a *query multinomial* based on a multi-tag query string, we can quickly retrieve relevant songs based on the Kullback-Liebler (KL) divergence between the query multinomial and all semantic multinomials in our database of automatically annotated tracks. The semantic multinomial representation is also useful for related audio information tasks such as ‘retrieval-by-semantic-similarity’ [Berenzweig et al. (2004); Barrington et al. (2007a)].

It should be noted that we use a very basic frame-based audio feature representation. We can imagine using alternative representations, such as those that attempt to model higher-level notions of harmony, rhythm, melody, and timbre. Similarly, our probabilistic SML model (a set of GMMs) is one of many models that have been de-

veloped for image annotation [Blei and Jordan (2003); Feng et al. (2004)]. Future work may involve adapting other models for the task of audio annotation and retrieval. In addition, one drawback of our current model is that, by using GMMs, we ignore all temporal dependencies between audio feature vectors. Future research will involve exploring models, such as hidden Markov models, that explicitly model the longer-term temporal aspects of music.

2.9 Acknowledgments

Chapter 2, in part, is a reprinted of material as it appears in the IEEE Transaction on Audio, Speech, and Language Processing, Turnbull, Douglas; Barrington, Luke; Torres, David; Lanckriet, Gert, February 2008. In addition, Chapter 2, in part, is a reprint of material as it appears in the ACM Special Interest Group on Information Retrieval, Turnbull, Douglas; Barrington, Luke; Torres, David; Lanckriet, Gert, July 2007. The dissertation author was the primary investigator and author of these papers.

Table 2.3: Music annotation results. Track-level models have $K = 8$ mixture components, tag-level models have $R = 16$ mixture components. A = annotation length (determined by the user), $|\mathcal{V}|$ = vocabulary size.

Category	$A / \mathcal{V} $	Model	Precision		Recall	
All Tags	10 / 174	Random	0.144	(0.004)	0.064	(0.002)
		Human	0.296	(0.008)	0.145	(0.003)
		UpperBnd	<i>0.712</i>	(0.007)	<i>0.375</i>	(0.006)
		ModelAvg	0.189	(0.007)	0.108	(0.009)
		MixHier	0.265	(0.007)	0.158	(0.006)
Emotion	4 / 36	Random	0.276	(0.012)	0.113	(0.004)
		Human	0.453	(0.014)	0.180	(0.006)
		UpperBnd	<i>0.957</i>	(0.005)	<i>0.396</i>	(0.010)
		ModelAvg	0.366	(0.012)	0.179	(0.005)
		MixHier	0.424	(0.008)	0.195	(0.004)
Genre	2 / 31	Random	0.055	(0.005)	0.079	(0.008)
		Human	0.268	(0.017)	0.290	(0.021)
		UpperBnd	<i>0.562</i>	(0.026)	<i>0.777</i>	(0.018)
		ModelAvg	0.122	(0.012)	0.161	(0.017)
		MixHier	0.171	(0.009)	0.242	(0.019)
Instrumentation	4 / 24	Random	0.141	(0.009)	0.195	(0.014)
		Human	0.416	(0.014)	0.522	(0.008)
		UpperBnd	<i>0.601</i>	(0.015)	<i>0.868</i>	(0.018)
		ModelAvg	0.267	(0.008)	0.320	(0.022)
		MixHier	0.259	(0.010)	0.381	(0.021)
Solo	1 / 9	Random	0.031	(0.007)	0.155	(0.035)
		Human	0.104	(0.020)	0.158	(0.034)
		UpperBnd	<i>0.197</i>	(0.019)	<i>0.760</i>	(0.052)
		ModelAvg	0.057	(0.012)	0.231	(0.033)
		MixHier	0.060	(0.012)	0.261	(0.050)
Usage	2 / 15	Random	0.073	(0.008)	0.154	(0.016)
		Human	0.125	(0.012)	0.175	(0.023)
		UpperBnd	<i>0.363</i>	(0.014)	<i>0.814</i>	(0.031)
		ModelAvg	0.103	(0.010)	0.170	(0.017)
		MixHier	0.122	(0.012)	0.264	(0.027)
Vocal	2 / 16	Random	0.062	(0.007)	0.153	(0.018)
		Human	0.188	(0.021)	0.304	(0.023)
		UpperBnd	<i>0.321</i>	(0.017)	<i>0.788</i>	(0.019)
		ModelAvg	0.102	(0.008)	0.226	(0.016)
		MixHier	0.134	(0.005)	0.335	(0.021)

Table 2.4: Sound effects annotation results. $A = 6$, $|\mathcal{V}| = 348$.

Model	Recall	Precision
Random	0.018 (0.002)	0.012 (0.001)
UpperBnd	0.973 (0.004)	0.447 (0.009)
ModelAvg ($K = 4$)	0.360 (0.014)	0.179 (0.010)
MixHier ($K = 8, R = 16$)	0.306 (0.010)	0.145 (0.005)

Table 2.5: Music retrieval results. $|\mathcal{V}| = 174$.

Category	$ \mathcal{V} $	Model	MeanAP	MeanAROC
All Tags	174	Random	0.231 (0.004)	0.503 (0.004)
		ModelAvg	0.372 (0.008)	0.682 (0.006)
		MixHier	0.390 (0.004)	0.710 (0.004)
Emotion	36	Random	0.327 (0.006)	0.504 (0.003)
		ModelAvg	0.486 (0.013)	0.685 (0.010)
		MixHier	0.506 (0.008)	0.710 (0.005)
Genre	31	Random	0.132 (0.005)	0.500 (0.005)
		ModelAvg	0.309 (0.020)	0.695 (0.008)
		MixHier	0.329 (0.012)	0.719 (0.005)
Instrumentation	24	Random	0.221 (0.007)	0.502 (0.004)
		ModelAvg	0.372 (0.015)	0.694 (0.008)
		MixHier	0.399 (0.018)	0.719 (0.006)
Solo	9	Random	0.106 (0.014)	0.502 (0.004)
		ModelAvg	0.190 (0.028)	0.688 (0.008)
		MixHier	0.180 (0.025)	0.712 (0.006)
Usage	15	Random	0.169 (0.012)	0.501 (0.005)
		ModelAvg	0.231 (0.012)	0.684 (0.007)
		MixHier	0.240 (0.016)	0.707 (0.004)
Vocal	16	Random	0.137 (0.006)	0.502 (0.004)
		ModelAvg	0.234 (0.019)	0.680 (0.007)
		MixHier	0.260 (0.018)	0.705 (0.005)

Table 2.6: Sound effects retrieval results. $|\mathcal{V}| = 348$.

Model	Mean AP	Mean AROC
Random	0.051 (0.002)	0.506 (0.004)
ModelAvg ($K = 4$)	0.183 (0.003)	0.785 (0.005)
MixHier ($K = 8, R = 16$)	0.331 (0.008)	0.784 (0.006)

Table 2.7: Qualitative music retrieval results for our SML model. Results are shown for 1-, 2- and 3-tag queries.

Query	Returned Songs
Pop	The Ronettes- Walking in the Rain The Go-Gos - Vacation Spice Girls - Stop Sylvester - You make me feel mighty real Boo Radleys - Wake Up Boo!
Female Lead Vocals	Alicia Keys - Fallin' Shakira - The One Christina Aguilera - Genie in a Bottle Junior Murvin - Police and Thieves Britney Spears - I'm a Slave 4 U
Tender	Crosby Stills and Nash - Guinnevere Jewel - Enter from the East Art Tatum - Willow Weep for Me John Lennon - Imagine Tom Waits - Time
Pop AND Female Lead Vocals	Britney Spears - I'm a Slave 4 U Buggles - Video Killed the Radio Star Christina Aguilera - Genie in a Bottle The Ronettes - Walking in the Rain Alicia Keys - Fallin'
Pop AND Tender	5th Dimension - One Less Bell to Answer Coldplay - Clocks Cat Power - He War Chantal Kreviazuk - Surrounded Alicia Keys - Fallin'
Female Lead Vocals AND Tender	Jewel - Enter from the East Evanescence - My Immortal Cowboy Junkies - Postcard Blues Everly Brothers - Take a Message to Mary Sheryl Crow - I Shall Believe
Pop AND Female Lead Vocals AND Tender	Shakira - The One Alicia Keys - Fallin' Evanescence - My Immortal Chantal Kreviazuk - Surrounded Dionne Warwick - Walk on by

Table 2.8: Music retrieval results for 1-, 2-, and 3-tag queries. See Table 2.3 for SML model parameters.

Query Length	Model	MeanAP	MeanAROC
1-tag (159/159)	Random	0.173	0.500
	SML	0.307	0.705
2-tags (4,658/15,225)	Random	0.076	0.500
	SML	0.164	0.723
3-tags (50,471/1,756,124)	Random	0.051	0.500
	SML	0.120	0.730

Chapter 3

Using a Game to Collect Tags for Music

3.1 Introduction

Collecting high-quality semantic annotations of music is a difficult and time-consuming task. Examples of annotations include chorus onset times [Goto (2006a)], genre labels [Tzanetakis and Cook (2002)], and music similarity matrices [Pampalk et al. (2005)]. In recent years, the Music Information Retrieval (MIR) community has focused on collecting standard data sets of such annotations for the purpose of system evaluation (e.g., MIREX competitions [Downie (2007)], RWC Database [Goto (2004)]). These data sets are relatively small, however, when compared to other domain specific data sets for speech recognition [Garofolo et al. (1993)], computer vision [Carneiro et al. (2007)], and natural language processing [Lewis (1997); Roukos et al. (1995)].

Traditionally, one amasses annotations through hand-labeling of music [Goto (2006a); Tzanetakis and Cook (2002)], conducting surveys [Pandora¹, Moodlogic², Turnbull et al. (2007a)], and text-mining web documents [Turnbull et al. (2006); Knees et al. (2008); Whitman and Ellis (2004)]. Unfortunately, each approach has drawbacks – the first two methods do not scale since they are time consuming and costly; the third generally produces results that are inconsistent with true semantic information.

¹www.pandora.com

²www.moodlogic.com

To collect high quality data en masse for very low cost, we propose the use of web-based games as our annotation engine. Recently, von Ahn et. al. created a suite of games (ESP Game [von Ahn and Dabbish (2004a)], Peekaboom [von Ahn (2006)], Phetch [von Ahn et al. (2006)]) for collecting semantic information about images. On the surface, these ‘games with a purpose’ present a platform for user competition and collaboration, but as a side effect they also provide data that one can distill into a useful form. This technique is called *human computation* because it harnesses the collective intelligence of a large number of human participants. Through this game-based approach, a population of users can solve a large problem (i.e., labeling all the images on the Internet) by the contributions of individuals in small groups (i.e., labeling a single image.)

In this paper, we describe the *Listen Game*, a multi-player, web-based game designed to collect associations between audio content and words. Listen game is designed with the notion that ‘music is subjective’. That is, players will often disagree on the words that describe a song. By collecting a votes from a large number of players, we democratically represent song-word relationships as real-valued *semantic weights* that reflect a strength of association, rather than all-or-nothing binary labels. The *initial* vocabulary consists of preselected ‘musically relevant’ words, such as those related to musical genre, instrumentation, or emotional content. Over time, the game has the ability to *grow* the vocabulary of words by recording the responses of players during special modes of play.

While one can think of the Listen Game as an entertaining interface for collaboration and competition, we will show that it is also a powerful tool for collecting semantic music information. In previous work [Turnbull et al. (2007a)], we presented a system that can automatically both *annotate* novel music with semantically meaningful words and *retrieve* relevant songs from a large database. Our system learns a supervised multi-class labeling (SML) model [Carneiro et al. (2007)] using a heterogeneous data set of audio content and semantic annotations; while previous human computation research evaluates performance based on annotation accuracy through user studies, we

use the data to train a machine learning system which, in turn, can annotate novel songs.

3.2 Collecting Music Annotations

A supervised learning approach to semantic music annotation and retrieval requires that we have a large corpus of song-word associations. Early work in music classification (by genre [Tzanetakis and Cook (2002); McKinney and Breebaart (2003)], emotion [Li and Ogihara (2003)], instrument [Essid et al. (2005)]) either used music corpora hand-labeled by the authors or made use of existing song metadata. While hand-labeling generally results in high quality labels, it does not scale easily to hundreds of labels per song over thousands of songs. To circumvent the hand-labeling bottleneck, companies such as Pandora employ dozens of musical experts whose full-time job is to tag songs with a large vocabulary of musically relevant words. Unfortunately, the administrators at Pandora have little incentive to make their data publicly available³.

In [Whitman and Ellis (2004)], Whitman and Ellis propose crawling the Internet to collect a large number of web-documents and summarizing their content using text-mining techniques. From web-documents associated with *artists*, they could learn binary classifiers for musically relevant words by associating those words with the artists' songs. In previous work [Turnbull et al. (2006)], we mined music reviews associated with *songs* and demonstrated that we could learn a supervised multi-class labeling (SML) model over a large vocabulary of words. While web-mining is a more scaleable approach than hand-labeling, we found through informal experiments that the data collected was of low-quality, in that extracted words did not necessarily provide a good description of a song. This is due to the fact that, in general, authors of web-documents do not explicitly make decisions about the relevance of given word when writing about songs and/or artists.

A third approach uses surveys to collect semantic information about music. Moodlogic allows their customers to annotate music using a standard survey contain-

³based on personal discussions with Pandora founder Tim Westergren

ing questions about genre, instrumentation, emotional characteristics, etc. Because this data is not publicly available, we created a data set of songs and semantic word associations ourselves. The result is the CAL500 data set of 500 songs, each of which has been annotated using a vocabulary of 173 words by a minimum of three people. Data collection took over 200 person-hours of human test, and resulted in approximately 261,000 individual word-song associations. This approach did result in higher quality song-word associations than the web data [Turnbull et al. (2006)], but required that we pay test subjects for their time. The more problematic issue, however, is that surveys are tedious; despite financial motivation, test subjects become quickly tire of lengthy surveys, resulting in inaccurate annotations.

The idea of just *asking* people in the style of a survey is not new. The Open Mind Initiative [Stork (2000)], for example, seeks to gather general knowledge for computers. As said before, however, people do not often have the proper motivation to aid a data collection effort. Recently, von Ahn et al introduce *human computation* as a promising alternative to traditional surveys. A progression of three web-based games ([von Ahn and Dabbish (2004a); von Ahn (2006); von Ahn et al. (2006)]) demonstrates the concept of using humans, rather than machines, to perform the critical computations of the system *that computers cannot yet do*.

Human computation games have the property that players generate reliable annotations based on incentives built into the game. For example, the ESP Game [von Ahn and Dabbish (2004a)] was developed to collect reliable word-image pairs. In it, the game client shows the same image to pairs of players and asks each player to ‘enter what your partner is thinking.’ Invariably, since they have no means of communicating, the words they enter have something to do with the image. Since two people *independently* suggest the same word to describe the image, the game mechanisms ensure annotation quality.

Human computation, in its game manifestation, also addresses the issue of collecting *lots* of annotations by turning annotation into an entertaining task. The ESP Game has gathered over 10 Million word-image associations. Games have the advan-

tage that they can build a sense of community and loyalty in users; statistics from [von Ahn and Dabbish (2004a)] highlight that some people have played in multiple 40 hour per week spans. Since they require little maintenance and run 24-hours per day; a game can constantly collect new information from multiple players.

Developing annotation games for music is a natural progression from earlier work with images since: 1) There is a demand for semantic information about music; 2) People enjoy talking about, sharing, discovering, arguing about, and listening to music. We have designed and implemented the Listen Game specifically with these ideas in mind. At present, Mandel and Ellis have independently conceived of and proposed another game, MajorMinor. In particular, their game asks the user to listen to a clip from a song and type words that describe it. The individual receives points in an offline manner if another individual enters the same word at a previous or future point in time. The Bee-Watcher-Watcher watched the Bee-Watcher. We consider Major Minor conceptually similar to the Open Mind Initiative [Stork (2000)], and less like a human computation game, because of the open-ended data entry format, as well as the lack of real-time interaction. However, both Listen Game and MajorMinor are tools that collect reliable song-word associations and allow users to suggest new words to describe music.

3.3 The Listen Game

When designing a human computation game for music, it is important to understand that music is ‘inherently subjective’. To this end, we have tried to create a game that is collaborative in nature so that users share their opinion, rather than be judged as correct or incorrect. Data collection also reflects this principle in that, we are interested in collecting the strength of associations between a word and a song, rather than an all or nothing relationship (i.e., a binary label). Image annotation, on the other hand, often involves binary relationships between an image and the objects (‘sailboat’), scene information (‘landscape’), and visual characteristics (‘red’) represented.

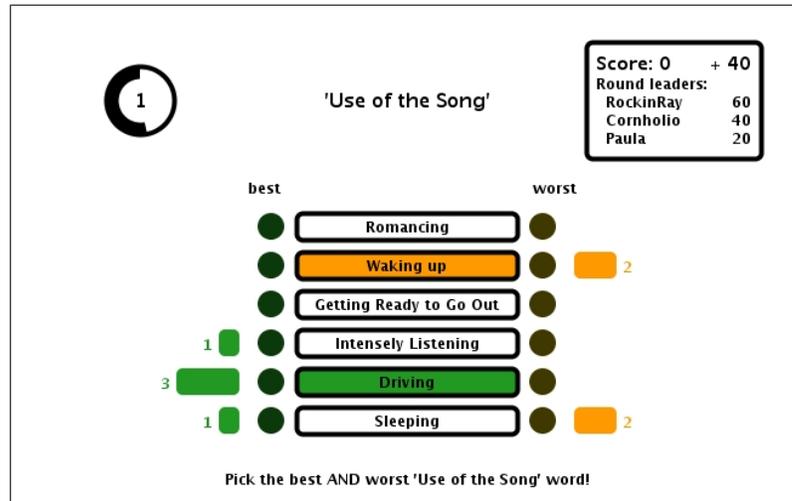


Figure 3.1: Normal Round: players select the best word and worst word that describes the song.

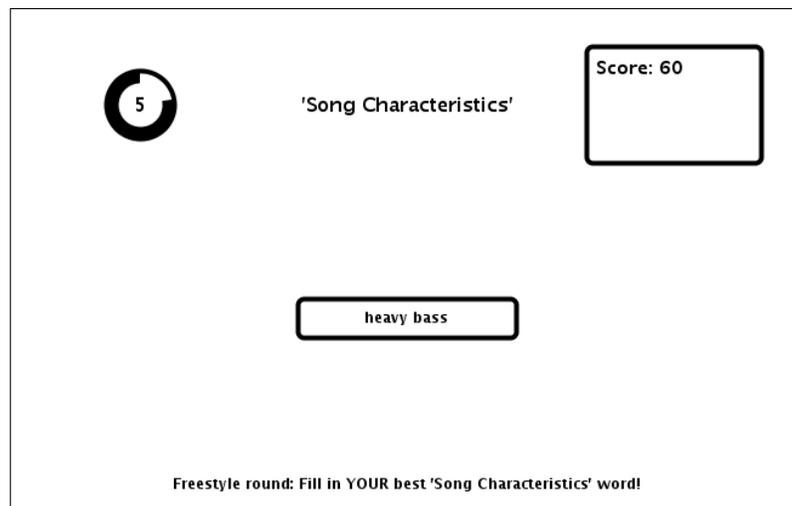


Figure 3.2: Freestyle Round: players enter a word that describes the song.

3.3.1 Description of Gameplay

Listen game is a round-based game where a player play for 8 consecutive rounds. In a regular round (Figure 3.1), the game server selects a clip (~ 15 seconds in duration) and six words associated with a semantic category (e.g., Instrumentation, Usage, Genre). The game client plays the clip and displays the category and word choices in a randomly

permuted order (to avoid order bias). The player then selects the best word describing the clip, as well as the worst word to describe the clip. Once she has fully committed her choices, the game client then displays the voting results by all the other players. We calculate a player’s score S by

$$S = 100 * (\text{fraction in agreement for best}) \\ + 100 * (\text{fraction in agreement for worst}).$$

In a *freestyle* round (Figure 3.2), the game client plays a *preview* clip and displays a category representing regular play for the *next* round. It then asks the player to enter an appropriate description for the song. The game server then incorporates her entry into the next round. Upon finishing her 8 rounds, the player then transitions to the summary panel. Here, the client displays her score over those rounds, the songs played, and various game statistics.

The game also includes a high scoreboard, personal profile page, and ‘fun fact’ widgets to display other interesting information (e.g., ‘The most “Annoying” song of the day’). Personal information is recorded (given on a voluntary basis) so that we can collect demographically-specific semantic information about music.

3.3.2 Quality of Data

Note that although individual best-worst choices by players are binary, but the aggregate song-word associations are not binary; rather, one may interpret them as real-valued weights proportional to the percentage of players who agree that a word does (or does not) describe a song.

We calculate the semantic weight w as a function of positive (‘Best’), negative votes (‘Worst’), and potential votes (the number of times a song-word pair is presented to a user):

$$w = \begin{cases} 0, & \text{if } \#(\text{Positives}) - \#(\text{Negatives}) < 2 \\ w', & \text{otherwise} \end{cases}$$

$$w' = \max \left(0, \left[\frac{\#(\text{Positives}) - \#(\text{Negatives})}{\#(\text{Potential Votes})} \right] \right).$$

For a song-word pair to be reliable, we require that at least two people make the association in any given round. We would hope that with more data, we could raise the threshold for agreement significantly.

Admittedly, the policy is heuristic and doesn't take into account the full information that collected by our game. That is, from a generative perspective, we could consider each round of a game as a draw from a multinomial distribution over a subset of words in our vocabulary. Ultimately, we would like to model some words as Bernoulli distributions ('Electric Guitar', 'Breathy Vocals') and other sets of words as multinomial distributions over the entire vocabulary ('Emotions', 'Usages'). While we plan to explore these ideas in future research, training of our SML model requires only that we identify a set of positively associated song for each word in the vocabulary. We need not learn 'negative class' models as would be the case if we were to use a binary classifier (e.g., Support Vector Machine) for each word.

3.4 Supervised Multiclass Labeling Model

We use the semantic song-word associations collected using Listen Game to train our supervised multi-class labeling (SML) model. The SML model was originally developed by Carneiro et al. [Carneiro et al. (2007)] for the tasks of image annotation and retrieval. First, each image is represented as a 'bag' of feature vectors. Then, for each word in a pre-defined vocabulary, we learn a Gaussian mixture model (GMM) over the feature space using the images associated with that word. We estimate these 'word-level' GMMs by combining 'song-level' GMMs (one trained on the feature vectors extracted from a single image) using the mixture hierarchies expectation-maximization algorithm (MH-EM for short) [Vasconcelos (2001)]. MH-EM has a number of advantages over more traditional parameter estimation techniques (e.g., direct estimation and model averaging) that include beneficial regularization and a reduction in computation time in

terms of both parameter estimation and inference.

In [Turnbull et al. (2007a)], we showed how to use the SML model to annotate and retrieve songs using the CAL500 data set. We also extended MH-EM to allow for real-valued *semantic weights*, rather than binary labels. While using binary labels is quite natural for images where the majority of words are associated with ‘objective’ semantic concepts, music is more ‘subjective’ in that two listeners may not always agree a song is representative of a certain genre or emotion. The Listen Game directly reflects such a notion – we let a large group of users vote on the best and worst words to describe a song. Using our *weighted* MH-EM algorithm, we learn a GMM’s that reflect the strength of the semantic associations between that words and songs. We refer the reader to [Turnbull et al. (2007a)] for a full explanation of this algorithm, as well as other details related to audio feature extraction and semantic representation.

3.5 Evaluation of Listen Game Data

We show that the data collected using Listen Game, referred to as ‘Listen250’, is useful for automatic music annotation and retrieval. To accomplish this goal, we will then train an SML model using the Listen250 data and evaluate that model using the CAL500 data.

3.5.1 Cal500 and Listen250 Data

As mentioned in Section 3.2, we have previously collected the CAL500 data set that contains semantic weights between 500 songs (by 500 unique artists) and 174 words. The 174 words are part of a hierarchical vocabulary with six high-level semantic categories: genre, emotion, instrumentation, vocal characteristic, general song characteristics, and usage. We determine the ‘strength of association’ for these 87,000 word-song pairs by averaging the response of multiple individuals who each annotate the song using a standard survey [Turnbull et al. (2007a)].

More recently, we conducted a two-week pilot study of Listen Game. For clarity, we pared our vocabulary down to 120 words by eliminating ambiguous and less well known words. (For the experiments reported in Section 3.5.4, we again pare this vocabulary down to 82 words; we require that each word has been used to describe a minimum of five songs in the corpus.) To reduce the number of song-word pairs, we randomly selected a set of 250 songs from the CAL500 data set. Publicity for the game consisted of emails to the authors’ friends and family, a mass email to a Music-IR list, and word-of-mouth referrals.

During the two-week study, we collected 26,000 song-word labels (i.e., positive and negative votes) from 440 unique players. Twenty players played more than 30 8-round games, and five of them (including one of the authors) played more than 100 games. In addition, players generated 775 new words that were not part of the original 120-word CAL500 vocabulary. Some standouts include subgenres (‘Psychedelic’, ‘Lounge’), specific usages (‘Good for a Hangover’, ‘Cooking’), creative adjectives (‘Airy’, ‘Fun Loving’), and slang (‘Agro’, ‘Moshing’).

While Listen Game provides high-quality annotations for songs, a player must choose one word from a list of six words. This setup is not as ideal as a musical survey, since selecting from a set can introduce labeling problems. For example, if there are two relevant words, a player is forced to pick one word and thus reduce the semantic weight of the other. If there are no relevant words, a player must pick a bad word at random.

While we do not collect ideal data, the data is still valuable in that over time strong song-word associations will emerge. We can then use the strongest associations to train our SML model. Since the CAL500 data set is closer to our ideal ground truth, we use it to evaluate the performance of our SML model.

3.5.2 Qualitative Analysis

In Table 3.1, we present two annotations of two songs generated by humans and machines. We summarize the human annotations by ranking words (within each se-

Table 3.1: Musical Madlibs: annotations generated directly using the semantic weights that are created by Listen Game, and automatically generated annotations where the song is presented to the Listen250 SML model as novel audio content.

<p style="text-align: center;">Norah Jones - Don't Know Why Generated using Listen Game data</p> <p>This is cool jazz, soul song that is mellow and positive. It features female vocal, piano, bass, and breathy, aggressive vocals. It is a song light beat and with a catchy feel that you might like listen to while studying.</p> <p style="text-align: center;">Automatically Generated using SML model</p> <p>This is soft rock, jazz song that is mellow and sad. It features piano, synthesizer, ambient sounds, and monotone, breathy vocals. It is a song slow tempo and with low energy that you might like listen to while studying.</p>
<p style="text-align: center;">Rick James - Super Freak Generated using Listen Game data</p> <p>This is R&B, funk song that is positive and cheerful. It features male vocal, piano, acoustic guitar, and high-pitched, aggressive vocals. It is a song with a catchy feel and with a changing energy level that you might like listen to while at a party.</p> <p style="text-align: center;">Automatically Generated using SML model</p> <p>This is poppy, R&B song that is not mellow and cheerful. It features sequencer, synthesizer, male vocal, and spoken, rapping vocals. It is a song that is very danceable and with a synthesized texture that you might like listen to while at a party.</p>

mantic category) according to the semantic weights calculated from Listen Game votes. Note that while ‘Don’t know why’ by Norah Jones is labeled as both ‘Cool Jazz’ and ‘Soul’, the labels do not necessarily reflect the *best* genre to describe the song. Looking at the database of annotations, we see that ‘Cool Jazz’ label was selected by multiple people in a round where there were no truly relevant words. One could expect that after many rounds, however, that concentrated votes for relevant words would reduce the semantic weight assigned to words appearing in rounds with no clear choice. Adding an

‘abstain’ button to skip the song and an ‘audio dictionary’ may very well reduce errors as well.

The second set of annotations in Table 3.1 are those assigned by the SML model trained using CAL250 data. One can interpret the audio content as novel data used to test the SML model. While the above annotations do show that the CAL250 SML model produces semantically meaningful words, they are, on the whole, noticeably inferior to the annotations produced by the CAL500 SML model. To be fair, however, we consider the Listen250 data set to be sparse, since there have only been on average two ‘potential votes’ for each of the 20,500 song-word pairs.

3.5.3 Qualitative Evaluation

We use four standard information retrieval (IR) metrics to measure annotation and retrieval performance: mean per-word precision (pwPrecision) [Feng et al. (2004)], mean per-word recall (pwRecall) [Carneiro et al. (2007)], mean average precision (mean-AP) [Turnbull et al. (2007a)], and mean area under the receiver operating characteristic curve (meanAROC) [Turnbull et al. (2006)].

The pwPrecision and pwRecall metrics are tied to annotation performance. First, we annotate each song with a fixed number of words as picked by the SML model. For each word w in our vocabulary, $|w_H|$ is the number of songs that have word w in the “ground truth” annotation, $|w_A|$ is the number of songs that our model annotates with word w , and $|w_C|$ is the number of “correct” words that have been used both in the ground truth annotation and by the model. pwRecall is $|w_C|/|w_H|$ and pwPrecision is $|w_C|/|w_A|$. While trivial models can easily maximize one of these measures (e.g., by labeling all songs with a certain word or, instead, none of them), achieving excellent precision and recall simultaneously requires a truly valid model.

We use the meanAP and MeanAROC metrics to compare retrieval performance. We calculate the average precision (AP) by moving down our ranked list of test songs and averaging the precisions at every point where we correctly identify a new song. An

Table 3.2: Model Evaluation: The semantic information for CAL models was collected using a survey, while the Listen model was train using data collected using Listen Game. We annotate each song with 8 words.

Model	Annotation		Retrieval	
	pwRecall	pwPrecision	meanAP	meanAROC
Random	0.092	0.058	0.188	0.501
Listen250	0.188	0.289	0.368	0.661
CAL 250	0.215	0.333	0.410	0.701
CAL500	0.224	0.338	0.429	0.722

ROC curve is a plot of the true positive rate as a function of the false positive rate as we move down this ranked list of songs. The area under the ROC curve (AROC) is found by integrating the ROC curve and is upper bounded by 1.0. Random guessing in a retrieval task results in an AROC of 0.5. MeanAP and MeanAROC are found by averaging the AP and AROC across all words in our vocabulary.

3.5.4 Results

In Table 3.2, we compare the performance of three SML models: Listen250 trained using 225 songs and weighted using annotations collected with Listen Game, CAL250 trained using 225 songs weighted using responses to surveys, and CAL500 trained using 450 songs also weighted based on survey responses. We evaluate all of these model with the CAL500 data using 10-fold cross-validation. All difference are significant (paired t-test with $\alpha = 0.05$) with the exception of pwRecall and pwPrecision between CAL250 and CAL500.

As we might expect, the models (CAL250, CAL500) trained with survey data produces better annotation and retrieval performance than the model (Listen250) trained with game data. The model (CAL500) trained with using more songs performs better with respect to retrieval performance then the model (CAL500) with fewer songs.

We would expect the performance of all model, but especially Listen250, to im-

prove with both more training songs and better estimates of the word-song relationships. For example, we noticed an improvement in MeanAROC for Listen250 from 0.640 to 0.661 during the last 4 days of our two-week pilot study when after we had collected approximately 35% more data. At the end of our pilot study, we had shown each of our 20500 word-song pairs only twice to a player. We consider this far too few to gain a good estimate of the true strength of the relationship between a word and a song, especially considering that player were forced to choose a word from a list contain other potentially relevant words.

3.6 Discussion

We believe that human computation games could be a powerful tool for the Music Information Retrieval (MIR) community. Although, we plan to keep improving Listen Game, we have begun discussing ideas for other games. One specific idea is a game based on the idea of building music similarity matrices. We can also image games like ‘Dance Dance Revolution’ and ‘Guitar Hero’ as well as games that could be re-engineered and made into a tool that collects beat or note onset times. The challenge is to develop games that are both entertaining to players and useful for data collection.

During our two-week pilot study, we collected 26,000 semantic song-word pairs. This is still an order of magnitue less than then the 261,000 song-word pairs that we collected for the CAL500 dataset. To this end, it is not surprising that, at this point, the models trained with Listen Game data are inferior to CAL250 and CAL500 models. However, in the long run, we believe that Listen Game will be able to collect more data and produce superior models. To this end, we are also planning to develop statistically sound models to intepret the data provided by Listen Game in a less heuristic manner.

As our user base grows we will be able to spawn multiple simultaneous games and collect data at an increasing rate. These games can feature specific types of music (e.g., based on genre, emotion, usage, etc.) so that we can develop new vocabularies (using the freestyle round) and collect more focused song-word associations. Our next

step will be to improve Listen Game based on user feedback, by adding functionality, designing a better user interface, and conducting focused user studies. The best example of a suggestion is the idea to build a version of Listen Game for visually impaired users.

3.7 Acknowledgments

Chapter 3, in full, is a reprint of material as it appears in the International Conference on Music Information Retrieval, Turnbull, Douglas; Liu, Ruoran; Barrington, Luke; Lanckriet, Gert, September 2007. The dissertation author was the primary investigator and author of this papers.

Chapter 4

Comparing Approaches to Collecting Tags for Music

4.1 Introduction

Tags are text-based tokens, such as “happy”, “classic rock”, and “distorted electric guitar”, that can be used to annotate songs. They represent a rich source of semantic information that is useful for text-based music retrieval (e.g., Turnbull et al. (2008)), as well as recommendation, discovery, and visualization Lamere and Celma (2007). Tags can be collected from humans using surveys Turnbull et al. (2008); Clifford (2007), social tagging websites Levy and Sandler (2007), or music annotation games Turnbull et al. (2007c); Mandel and Ellis (2007); Law et al. (2007). They can also be generated by text mining web-documents Knees et al. (2007); Whitman and Ellis (2004) or by autotagging audio content Turnbull et al. (2008); Eck et al. (2007); Tzanetakis and Cook (2002). In Section 4.2, we introduce key concepts associated with tag collection, and use them to highlight the strengths and weaknesses of each of these five approaches. In Section 4.3, we describe one implementation of a system for each approach and evaluate its performance on a tag-based music retrieval task. In the final section, we describe a simple hybrid system that combines the output from each of our individual systems.

Table 4.1: Comparing the *costs* associated with five tag collection approaches: The bold font indicates a strength for an approach.

Approach	Cost / Scalability		
	Financial	Human	Computational
Survey Pandora CAL500	Expensive \$20-\$30 / survey ~ \$1 / survey	Expensive 20 min / survey 6 min / survey	Cheap Tools, DB Tools, DB
Social Tags LastFM	Moderate Support Popular Website	Moderate Lots of Users	Cheap Website, Plugins, DB
Game Listen Game	Moderate Design, Deploy, Promote Game	Moderate Trade Entertainment for Tags	Cheap Game Server, DB
Web Documents RS System	Cheap Fully Automated	None	Moderate Webcrawling, Text Processing
Autotags SML Model	Cheap Fully Automated	Cheap Training Data Set	Expensive Audio Processing & Modeling

4.2 Collecting Tags

In this section, we describe five approaches to collecting music tags. Three approaches (surveys, social tags, games) rely on human participation, and as such, are expensive in terms of financial cost and human labor. Two approaches (text mining, autotagging) rely on automatic methods that are computationally intense, but require less direct human involvement.

There are a number of key concepts to consider when comparing these approaches. The *cold start problem* refers to the fact songs that are not annotated cannot be retrieved. This problem is related to *popularity bias* in that popular songs (in the *short head*) tend to be annotated more thoroughly than unpopular songs (in the *long tail*) Lamere and Celma (2007). This often leads to a situation in which a short head song is ranked above a long tail song despite the fact that the long tail song may be more semantically relevant. We prefer an approach that avoids the cold start problem

Table 4.2: Comparing the *quality* of the tags collected using five tag collection approaches: The bold font indicates a strength for an approach.

Approach	Quality			
	'Cold Start'	Labeling	Vocabulary	Accuracy
Survey Pandora CAL500	Decent Long Backlog Small Sample Can Pick Songs	Strong	Structured, Fixed 400 Tags 174 Tags	Great Professional Redundancy
Social Tags LastFM	Poor Sparse in Long Tail	Weak	Unstructured, Extensible >960,000 Tags, lots of noise	Decent Adhoc Tagging
Game Listen Game	Decent Can Pick Songs	Depends on Game	Structured, Extensible 174 Tags	Good Competition, Redundancy
Web Documents RS System	Poor Sparse in Long Tail	Weak	Unstructured, Extensible Natural Language	Decent Noisy Documents
Autotags SML Model	Great Label all Songs w/ all Tags	Strong	Structured, Fixed Depends on Training Data	Decent Content-based analysis

(e.g., autotagging). If this is not possible, we prefer approaches in which we can explicitly control which songs are annotated (e.g., survey, games), rather than an approach in which only the more popular songs are annotated (e.g., social tags, web documents).

A *strong labeling* Carneiro et al. (2007) is when a song has been explicitly labeled or not labeled with a tag, depending on whether or not the tag is relevant. This is opposed to a *weak labeling* in which the absence of a tag from a song does not necessarily indicate that the tag is not relevant. For example, a song may feature drums but is not explicitly labeled with the tag “drum”. Weak labeling is a problem if we want to design a MIR system with high recall, or if our goal is to collect a training data set for a supervised autotagging system that uses discriminative classifiers (e.g., Eck et al. (2007); Whitman and Ellis (2004)).

It is also important to consider the size, structure, and extensibility of the tag vocabulary. In the context of text-based music retrieval, the ideal vocabulary is a large

and diverse set of semantic tags, where each tag describes some meaningful attribute or characterization of music. In this paper, we limit our focus to tags that can be used consistently by a large number of individuals when annotating novel songs based on the audio content alone. This does not include tags that are personal (e.g., “seen live”), judgmental (e.g., “horrible”), or represent external knowledge about the song (e.g., geographic origins of an artist). It should be noted that these tags are also useful for retrieval (and recommendation) and merit additional attention from the MIR community.

A tag vocabulary can be *fixed* or *extensible*, as well as *structured* or *unstructured*. For example, the tag vocabulary associated with a survey can be considered fixed and structured since the set of tags and the grouping of tags into coherent semantic categories (e.g., genres, instruments, emotions, usages) is predetermined by experts using domain knowledge Turnbull et al. (2008, 2007c). By contrast, social tagging communities produce a vocabulary that is extensible since any user can suggest any free-text token to describe music. This vocabulary is also unstructured since tags are not organized in any way. In general, we prefer an extensible vocabulary because a fixed vocabulary limits text-based retrieval to a small set of predetermined tags. In addition, a structured vocabulary is advantageous since the ontological relationships (e.g., genre hierarchies, families of instruments) between tags encode valuable semantic information that is useful for retrieval.

Finally, the accuracy with which tags are applied to songs is perhaps the most important point of comparison. Since there is no ideal ground truth and listeners do not always agree whether (or to what degree) a tag should be applied to a song (i.e., ‘the subjectivity problem’ McKay and Fujinaga (2006)), evaluating accuracy can be tricky. Intuitively, it is preferable to have trained musicologists, rather than untrained non-experts, annotate a music corpus. It is also advantageous to have multiple individuals, rather than a single person, annotate each song. Lastly, individuals who are given incentives to provide good annotations (e.g., a high score in a game) may provide better annotations than unmotivated individuals.

4.2.1 Conducting a Survey

Perhaps the most well-known example of the music annotation survey is Pandora's¹ "Music Genome Project" Clifford (2007); Westergren (2007). Pandora uses a team of approximately 50 expert music reviewers (each with a degree in music and 200 hours of training) to annotate songs using structured vocabularies of between 150-500 'musically objective' tags depending on the genre of the music Glaser et al. (2006). Tags, such as "Afro-Latin Roots", "Electric Piano Riffs" and "Political Lyrics", can be considered objective since, according to Pandora, there is a high level of inter-reviewer agreement when annotating the same song. Between 2000 and 2007, Pandora annotated over 600,000 songs Westergren (2007). Currently, each song takes between 20 to 30 minutes to annotate and approximately 15,000 new songs are annotated each month. While this labor-intensive approach results in high-quality annotations, Pandora must be very selective of which songs they choose to annotate given that there are already millions of songs by millions of artists².

Pandora, as well as companies like Moodlogic³ and All Media Guide (AMG)⁴, have devoted considerable amounts of money, time and human resources to annotate their music databases with high-quality tags. As such, they are unlikely to share this data with the MIR research community. To remedy this problem, we have collected the CAL500 data set of annotated music Turnbull et al. (2008). This data set contains one song from 500 unique artists each of which have been manually annotated by a minimum of three non-expert reviewers using a structured vocabulary of 174 tags. While this is a small data set, it is strongly labeled, relies on multiple reviews per song, and as such, can be used as a standard data set for training and/or evaluating tag-based music retrieval systems.

¹www.pandora.com

²In February 2008, Last.fm reported that their rapidly growing database consisted of 150 million songs by 16 million artists.

³<http://en.wikipedia.org/wiki/MoodLogic>

⁴www.allmusic.com

4.2.2 Harvesting Social Tags

Last.fm⁵ is a music discovery website that allows users to contribute *social* tags through a text box in their audio player interface. By the beginning of 2007, their large base of 20 million monthly users have built up an unstructured vocabulary of 960,000 free-text tags and used it to annotated millions of songs Miller et al. (2008). Unlike the Pandora and AMG, Last.fm makes much of this data available to the public through their Audiocrobbler⁶ site. While this data is a useful resource for the MIR community, Lamere and Celma Lamere and Celma (2007) point out a number of problems with social tags. First, there is often a sparsity of tags for new and obscure artists (cold start problem / popularity bias). Second, most tags are used to annotate artists rather than individual songs. This is problematic since we are interested in retrieving semantically relevant songs from eclectic artists. Third, individuals use ad-hoc techniques when annotating music. This is reflected by use of polysemous tags (e.g., “progressive”), tags that are misspelled or have multiple spellings (e.g., “hip hop”, “hip-hop”), tags used for self-organization (e.g., “seen live”), and tags that are nonsensical. Finally, the public interface allows for malicious behavior. For example, any individual or group of individuals can annotate an artist with a misleading tag.

4.2.3 Playing Annotation Games

At the 2007 ISMIR conference, music annotation games were presented for the first time: ListenGame Turnbull et al. (2007c), Tag-a-Tune Law et al. (2007), and MajorMiner Mandel and Ellis (2007). ListenGame is a real-time game where a large group of users is presented with a song and a list of tags. The players have to choose the best and worst tags for describing the song. When a large group of players agree on a tag, the song has a strong (positive or negative) association with the tag. This game, like a music survey, has the benefit of using a structured vocabulary of tags. It can be considered a strong labeling approach since it also collects information that reflects negative semantic

⁵www.last.fm

⁶<http://www.audioscrobbler.net/>

associations between tags and songs. Like the ESPGame for image tagging von Ahn and Dabbish (2004b), Tag-a-Tune is a two-player game where the players listen to a song and are asked to enter “free text” tags until they both enter the same tag. MajorMiner is similar in nature, except the tags entered by the player are compared against the database of previously collected tags in an offline manner. Like social tagging, the tags collected using both games result in a unstructured, extensible vocabulary.

A major problem with this game-based approach is that players will inevitably attempt to *game* the system. For example, the player may only contribute generic tags (e.g., “rock”, “guitar”) even if less common tags provide a better semantic description (e.g., “grunge”, “distorted electric guitar”). Also, despite the recent academic interest in music annotation games, no game has achieved large scale success. This reflects the fact that it is difficult to design a viral game for this inherently laborious task.

4.2.4 Mining Web Documents

Artist biographies, album reviews, and song reviews are another rich source of semantic information about music. There are a number of research-based MIR systems that collect such documents from the Internet by querying search engines Knees et al. (2008), monitoring MP3 blogs Celma et al. (2006), or crawling a music site Whitman and Ellis (2004). In all cases, Levy and Sandler point out that such web mined corpora can be *noisy* since some of the retrieved webpages will be irrelevant, and in addition, much of the text content on relevant webpages will be useless Levy and Sandler (2007).

Most of the proposed web mining systems use a set of one or more documents associated with a song and convert them into a single document vector (e.g., tf-idf representation) Knees et al. (2007); Whitman and Lawrence (2002). This *vector space* representation is then useful for a number of MIR tasks such as calculating music similarity Whitman and Lawrence (2002) and indexing content for a text-based music retrieval system Knees et al. (2007). More recently, Knees et. al. Knees et al. (2008) have proposed a promising new web mining technique called *relevance scoring* as an alternative to the vector space approaches. Both relevance scoring and vector space approaches

are subject to popularity bias since short head songs are generally represented by more documents than long tail songs.

4.2.5 Autotagging Audio Content

All previously described approaches require that a song be annotated by humans, and as such, are subject to the cold start problem. Content-based audio analysis is an alternative approach that avoids this problem. Early work on this topic focused (and continues to focus) on music classification by genre, emotion, and instrumentation (e.g., Tzanetakis and Cook (2002)). These classification systems effectively ‘tag’ music with class labels (e.g., ‘blues’, ‘sad’, ‘guitar’). More recently, *autotagging* systems have been developed to annotate music with a larger, more diverse vocabulary of (non-mutually exclusive) tags Turnbull et al. (2008); Eck et al. (2007); Sordo et al. (2007). In Turnbull et al. (2008), we describe a generative approach that learns a Gaussian mixture model (GMM) distribution over an audio feature space for each tag in the vocabulary. Eck et al. use a discriminative approach by learning a boosted decision stump classifier for each tag Eck et al. (2007). Finally, Sordo et al. present a non-parametric approach that uses a content-based measure of music similarity to propagate tags from annotated songs to similar songs that have not been annotated Sordo et al. (2007).

4.3 Comparing Sources of Tags

In this section, we describe one system for each of the tag collection approaches. Each has been implemented based on systems that have been recently developed within the MIR research community Turnbull et al. (2008); Knees et al. (2008); Turnbull et al. (2007c). Each produces a $|S| \times |T|$ *annotation matrix* \mathbf{X} where $|S|$ is the number of songs in our corpus and $|T|$ is the size of our tag vocabulary. Each cell $x_{s,t}$ of the matrix is proportional to the strength of semantic association between song s and tag t .

We set $x_{s,t} = \emptyset$ if the relationship between song s and tag t is missing (i.e., unknown). If the matrix \mathbf{X} has many empty cells, then we refer to the matrix as *sparse*, otherwise we refer to it as *dense*. Missing data results from both weak labeling and the

Table 4.3: Strengths and weaknesses of tag-based music annotation approaches

Approach	Strengths	Weaknesses
Survey	<ul style="list-style-type: none"> custom-tailored vocabulary high-quality annotations strong labeling control of which songs are annotated 	<ul style="list-style-type: none"> small, predetermined vocabulary human-labor intensive time consuming approach lacks scalability large existing databases are not shared
Social Tags	<ul style="list-style-type: none"> collective wisdom of crowds unlimited vocabulary provides social context large scale systems currently exist 	<ul style="list-style-type: none"> create & maintain popular social website ad-hoc annotation behavior, weak labeling sparse/missing in long tail weak labeling
Game	<ul style="list-style-type: none"> collective wisdom of crowds control over which songs are annotated fast paced for rapid data collection entertaining incentives produce high-quality annotation 	<ul style="list-style-type: none"> “gaming” the system listening to clips, rather than songs difficult to create viral gaming experience no large scale success to date
Web Docs	<ul style="list-style-type: none"> no direct human involvement provides social context large, publicly-available corpus of music-related documents 	<ul style="list-style-type: none"> noisy annotations due to text-mining sparse/missing in long tail weak labeling
Autotags	<ul style="list-style-type: none"> not affected by cold-start problem no direct human involvement strong labeling 	<ul style="list-style-type: none"> computationally intensive limited by training data based solely on audio content

cold start problem. Sparsity is reflected by the *tag density* of a matrix which is defined as the percentage of non-empty elements of a matrix.

Our goal is to find a tagging system that is able to accurately retrieve (i.e., rank-order) songs for a diverse set of tags (e.g., emotions, genres, instruments, usages). We quantitatively evaluate music retrieval performance of system a by comparing the matrix \mathbf{X}^a against the CAL500 matrix $\mathbf{X}^{\text{CAL500}}$ (see Section 4.2.1). The $\mathbf{X}^{\text{CAL500}}$ matrix is a binary matrix where $x_{s,t} = 1$ if 80% of the individuals annotate song s with tag t , and 0 otherwise (see Section V.a of Turnbull et al. (2008) for details). For the experiments reported in this section, we use a subset of 109 of the original 174 tags.⁷ We will assume that the subset of 87 songs from the Magnatunes Downie (2005) collection that are included in the CAL500 data set are representative of long tail music. As such, we can use this subset to gauge how the various tagging approaches are affected by popularity bias.⁸

Each system is compared to the CAL500 data set using a number of standard information retrieval (IR) evaluation metrics Knees et al. (2008): area under the receiver operation characteristic curve (AROC), average precision, R-precision, and Top-10 precision. An ROC curve is a plot of the true positive rate as a function of the false positive rate as we move down this ranked list of songs. The area under the ROC curve (AROC) is found by integrating the ROC curve and is upper-bounded by 1.0. A random ranking of songs will produce an expected AROC score of 0.5. Average precision is found by moving down our ranked list of test songs and averaging the precisions at every point where we correctly identify a relevant song. R-Precision is the precision of the top R -ranked songs where R is the total number of songs in the ground truth that have been annotated with a given tag. Top-10 precision is the precision after we have retrieved the top 10 songs for a given tag. This metric is designed to reflect the 10 items that would be displayed on the first results page of a standard Internet search engine.

⁷We have merged genre-best tags with genre tags, removed instrument-solo tags, removed some redundant emotion tags, and pruned other tags that are used to annotate less than 2% of the songs. For a complete list of tags, see <http://cosmal.ucsd.edu/cal>.

⁸It should be noted that 87 songs is a small sample.

Each value in Table 4.4 is the mean of a metric after averaging over all 109 tags in our vocabulary. That is, for each tag, we rank-order our 500 song data set and calculate the value of the metric using CAL500 data as our ground truth. We then compute the average of the metric using the 109 values from the 109 rankings.

4.3.1 Social Tags: Last.fm

For each of our 500 songs, we attempt to collect two lists of social tags from the Last.fm Audioscobbler website. One list is related specifically to the song and the other list is related to the artist. For the song list, each tag has a score ($x_{s,t}^{\text{Last.fm.Song}}$) that ranges from 0 (low) to 100 (high) and is a secret function (i.e., trade secret of Last.fm) of both the number and diversity of users who have annotated song s with tag t . For the artist list, the tag score ($x_{s,t}^{\text{Last.fm.Artist}}$) is again a secret function that ranges between 0 and 100, and reflects both tags that have been used to annotate the artist or songs by the artist. We found one or more tags for 393 and 472 of our songs and artists, respectively. This included at least one occurrence of 71 and 78 of the 109 tags in our vocabulary. While this suggests decent coverage, tag densities of 4.6% and 11.8%, respectively, indicate that the annotation matrices, $X^{\text{Last.fm.Song}}$ and $X^{\text{Last.fm.Artist}}$, are sparse even when we consider mostly short head songs. These sparse matrices achieve AROC of 0.57 and 0.58.

To remedy this problem, we create a single Last.fm annotation matrix by leveraging the Last.fm data in three ways. First, we match tags to their synonyms.⁹ For example, a song is considered to be annotated with ‘down tempo’ if it has instead been annotated with ‘slow beat’. Second, we allow wildcard matches for each tag. That is, if a tag appears as a substring in another tag, we consider it to be a wildcard match. For example, “blues” matches with “delta electric blues”, “blues blues blues”, “rhythm & blues”. Although synonyms and wildcard matches add noise, they increase the respective densities to 8.6% and 18.9% and AROC performance to 0.59 and 0.59. Third, we

⁹Synonyms are determined by the author using a thesaurus and by exploring the Last.fm tag vocabulary.

combine the song and artist annotation matrices in one annotation matrix:

$$\mathbf{X}^{\text{Last.fm}} = \mathbf{X}^{\text{Last.fm_Song}} + \mathbf{X}^{\text{Last.fm_Artist}}.$$

This results in a single annotation matrix that has a density of 23% and AROC of 0.62. 95 of the 109 tags are represented at least once in this matrix. However, the density for the Magnatunes (e.g., long tail) songs is only 3% and produces retrieval results that are not much better than random.

4.3.2 Games: ListenGame

In Turnbull et al. (2007c), Turnbull et al. describe a music annotation game called ListenGame in which a community of players listen to a song and are presented with a set of tags. Each player is asked to vote for the single *best* tag and single *worst* tag to describe the music. From the game, we obtain the annotation matrix \mathbf{X}^{Game} by letting

$$[\mathbf{X}^{\text{Game}}]_{s,t} = \#(\text{best votes}) - \#(\text{worst votes})$$

when song s and tag t are presented to the players.

During a two-week pilot study, 16,500 annotations (best and worst votes) were collected for a random subset of 250 CAL500 songs. Each of the 27,250 song-tag pairs were presented to users an average of 1.8 times. Although this represents a very small sample size, the mean AROC for the subset of 250 songs averaged over the 109-tag vocabulary is 0.65. Long tail and short head results do not accurately reflect the real-world effect of popularity bias since all songs were selected for annotation with equal probability. As such, these results have been omitted.

4.3.3 Web Documents: Weight-based Relevance Scoring

In order to extract tags from a corpus of web documents, we adapt the relevance scoring (RS) algorithm that has recently been proposed by Knees et al. Knees et al. (2008). They have shown this method to be superior to algorithms based on vector space representations. To generate tags for a set of songs, the RS works as follows:

1. **Collect Document Corpus:** For each song, repeatedly query a search engine with each song title, artist name, or album title. Collect web documents in search results. Retain the (many-to-many) mapping between songs and documents.
2. **Tag Songs:** For each tag
 - (a) Use the tag as a query string to find the relevant documents, each with an associated *relevance weight* (defined below) from the corpus.
 - (b) For each song, sum the relevance scores for all the documents that are related to the song.

We modify this algorithm in two ways. First, the relevance score in Knees et al. (2008) is inversely proportional to the rank of the relevant document. We use a weight-based approach to relevance scoring (WRS). The relevance weight of a document given a tag can be a function of the number of times the tag appears in the document (tag-frequency), the number of documents with the tag (document frequency), the number of total words in the document, the number of words or documents in the corpus, etc. For our system, the relevance weights are determined by the MySQL match function.¹⁰

We calculate an entry of the annotation matrix \mathbf{X}^{WRS} as,

$$\mathbf{X}_{s,t}^{\text{WRS}} = \sum_{d \in D_t} w_{d,t} I_{d,s}$$

where D_t is the set of relevant documents for tag t , $w_{d,t}$ is the relevance weight for document d and tag t , and $I_{d,s}$ is an indicator variable that is 1 if document d was found when querying the search engine with song s (in Step 1) and 0 otherwise. We find that weight-based RS (WRS) produces a small increase in performance over rank-based RS (RRS) (AROC of 0.66 vs. 0.65). In addition, we believe that WRS will scale better since the relevance weights are independent of the number of documents in our corpus.

The second modification is that we use *site-specific* queries when creating our corpus of web documents (Step 1). That is, Knees et. al. collect the top 100 documents returned by Google when given queries of the form:

¹⁰<http://dev.mysql.com/doc/refman/5.0/en/fulltext-natural-language.html>

- “<artist name>” music
- “<artist name>” “<album name>” music review
- “<artist name> ” “<song name>” music review

for each song in the data set. Based on an informal study of the top 100 webpages returned by non-site-specific queries, we find that many pages contain information that is only slightly relevant (e.g., music commerce site, ticket resellers, noisy discussion boards, generic biographical information). By searching music-specific sites, we are more likely to find detailed music reviews and in-depth artist biographies. In addition, the webpages at sites like Pandora and AMG All Music specifically contain useful tags in addition to natural language content.

We use site-specific queries by appending the substring ‘site:<music site url>’ to the three query templates, where <music site url> is the url for a music website that is known to have high quality information about songs, albums or artists. These sites include allmusic.com, amazon.com, bbc.co.uk, billboard.com, epinions.com, musicomh.com, pandora.com, pitchforkmedia.com, rollingstone.com, wikipedia.org. For these 10 music sites and one non-site-specific query, we collect and store the top 10 pages returned by the Google search engine. This results in a maximum of 33 queries and a maximum of 330 pages per song. On average, we are only able to collect 150 webpages per song since some of the long tail songs are not well represented by these music sites.

Our *site-specific weight-based relevance scoring* (SS-WRS) approach produces a relatively dense annotation matrix (46%) compared with the approach involving Last.fm tags. However, like the Last.fm approach, the density of the annotation matrix is greatly reduced (25%) when we consider only long tail songs.

4.3.4 Autotagging: Supervised Multiclass Labeling

In Turnbull et al. (2008), we use a supervised multiclass labeling (SML) model to automatically annotate songs with a diverse set of tags based on audio content analysis.

The SML model is parameterized by one Gaussian mixture model (GMM) distribution over an audio feature space for each tag in the vocabulary. The parameters for the set of GMMs are trained using annotated training data. Given a novel audio track, audio features are extracted and their likelihood is evaluated using each of the GMMs. The result is a vector of probabilities that, when normalized, can be interpreted as the parameters of a multinomial distribution over the tag vocabulary. This *semantic multinomial* distribution represents a compact and interpretable index for a song where the large parameter values correspond to the most likely tags.

Using 10-fold cross validation, we can estimate a semantic multinomial for each of the CAL500 songs. By stacking the 50 test set multinomials from each of the 10 folds, we can construct a strongly-labeled annotation matrix \mathbf{X}^{SML} that is based purely on the audio content. As such, this annotation matrix is dense and not affected by the cold start problem.

4.3.5 Summary

Comparing systems using a two-tailed, paired t-test ($N = 109$, $\alpha = 0.05$) on the AROC metric, we find that all pairs of the four systems are significantly different, with the exception of Game and Web Documents.¹¹ If we compare the systems using the other three metrics (Average Precision, R-Precision, and Top 10 Precision), we no longer find statistically significant differences. It is interesting that Social Tags and Web Documents (0.37) have slightly better Top 10 precision than Autotags (0.33). This reflects the fact that for some of the more common individual tags, we find that Social Tags and Web Documents have exceptional precision at low recall levels. For both Web Documents and Social Tags, we find significant improvement in retrieval performance of short head songs over long tail songs. However, as expected, there is no difference for Autotags. This confirms the intuition that systems based on web documents and social tags are influenced by popularity bias, whereas content-based autotagging systems are not.

¹¹Note that when we compare each system with the Game system, we compare both systems using the reduced set of 250 songs.

Table 4.4: Tag-based music retrieval: Each approach is compared using all *CAL500* songs and a subset of 87 more obscure *long tail* songs from the Magnatunes dataset. *Tag Density* represents the proportion of song-tag pairs that have a non-empty value. The four evaluation metrics (*AROC*, *Average Precision*, *R-Precision*, *Top-10 Precision*) are found by averaging over 109 tag queries. †Note that ListenGame is evaluated using half of the CAL500 songs and that the results do not reflect the realistic effect of the popularity bias (see Section 4.3.2).

Approach	Songs	Tag Density	AROC	Avg. Prec	R-Prec	Top10 Prec
Survey	All Songs	1.00	1.00	1.00	1.00	0.97
	Ground Truth	Long Tail	1.00	1.00	1.00	0.57
Baseline	All Songs	1.00	0.50	0.15	0.14	0.13
	Random	Long Tail	1.00	0.50	0.18	0.12
Social Tags	All Songs	0.23	0.62	0.28	0.30	0.37
	Last.fm	Long Tail	0.03	0.54	0.24	0.19
Game	All Songs	0.37	0.65	0.28	0.28	0.32
Web Docs	All Songs	0.67	0.66	0.29	0.29	0.37
	SS-WRS	Long Tail	0.25	0.56	0.25	0.18
Autotags	All Songs	1.00	0.69	0.29	0.29	0.33
	SML	Long Tail	1.00	0.70	0.34	0.30

4.4 Acknowledgments

Chapter 4, in full, is a reprint of material as it appears in International Conference on Music Information Retrieval, Turnbull, Douglas; Barrington, Luke; Lanckriet, Gert. September 2007. The dissertation author was the primary investigator and author of this papers.

Chapter 5

Combining Multiple Data Sources for Semantic Music Discovery

Individuals often use words to describe music. For example, one might say that “Wild Horses” by the Rolling Stones is “a sad folk-rock tune that features somber strumming of an acoustic guitar and a minimalist use of piano and electric slide guitar.” Such descriptions are full of semantic information that is useful for music discovery. Specifically, we can annotate music with *tags*, which are short text-based tokens, such as “sad”, “folk-rock”, and “electric slide guitar.” Once annotated, songs can be retrieved from a large database of music given a text-based query.

As discussed in Chapter 4, tags for music can be obtained from a variety of sources. For example, music tags can be *collected* from humans using surveys, social tagging websites or annotation games Turnbull et al. (2007c); Mandel and Ellis (2007). In addition, these tags can be generated automatically through an content-based audio analysis Eck et al. (2007); Turnbull et al. (2008) or by text-mining associated web documents Knees et al. (2008). Taken together, these sources provide a description of the acoustic content and place the music into a social context, both of which are important for music discovery.

In this paper, we describe four sources from which we collect music information.

Specifically, we use two representations of the audio content, one related to timbre and one related to harmony, and two more socially situated representations, one based on social tags and one based on web documents. While the audio representations are *dense*, the social representations are considered *sparse* since the strength of association between some songs and some tags is unknown (i.e., missing). It should be noted that less popular songs tend to be more sparse since fewer humans have annotated these songs (i.e., the "cold start" problem).

We then describe and compare three algorithms that combine these four complementary representations: calibrated score averaging, RankBoost Freund et al. (2003), and the combined-kernel SVM Lanckriet et al. (2004). The first two approaches are similar in that they combine sets of ranked orderings, where each ranking comes from each of the representations. They differ in how they deal with missing data and how they combine the rankings. For the third algorithm, we first design a kernel matrix for each representation. We then learn an optimal linear combination of the kernel matrices using convex optimization to produce a single "combined" kernel which can be used by a support vector machine (SVM) to rank order test set songs.

In the following subsection, we describe related work on acoustic and social music representations, as well as existing approaches to combining representations. In Section 5.1, we describe two acoustic and two social sources of music information. In Section 5.2, we describe our three algorithms for combining sources of music information. In Section 5.3, we provide a comparative analysis of these algorithms and contrast them to approaches that only use the individual music information sources.

5.0.1 Related Work

McKinney and Breebaart McKinney and Breebaart (2003) use 4 different feature sets to represent audio content and evaluate their individual performance on general audio classification and 7-way genre classification. They determine that features based on the temporal modulation envelope of low-level spectral features are most useful for these

tasks but suggest that intelligent combinations of features might improve performance. Tzanetakis and Cook Tzanetakis and Cook (2002) present a number of content-based features and concatenate them to produce a single vector to represent each song. They use these music feature vectors with standard classifiers (e.g., nearest neighbor, GMM) for the task of genre classification.

Knees et al. Knees et al. (2008) use semantic data mined from the results of web-searches for songs, albums and artists to generate a *contextual* description of the music based on large-scale, social input, rather than features that describe the audio *content*. Using this context data alone achieves retrieval results comparable to the best content-based methods. Whitman and Ellis Whitman and Ellis (2004) also leverage web-mined record reviews to develop an unbiased music annotation system.

Flexer et al. Flexer et al. (2006) combine information from two feature sources: tempo and MFCCs. They use a nearest-neighbor classifier on each feature space to determine an independent class-conditional probability for each genre, given each feature set. Using a naive Bayesian combination, they multiply these two probabilities and find that the resulting probability improves 8-way genre classification of dance music.

Lanckriet et al. Lanckriet et al. (2004) propose a more sophisticated method for combining information from multiple feature sources. They use a kernel matrix for *each* feature set to summarize similarity between data points and demonstrate that it is possible to learn a linear combination of these kernels that optimizes performance on a discriminative classification task. It has been shown that, for protein classification Lanckriet et al. (2004) and music retrieval Barrington et al. (2008) tasks, an optimal combination of heterogenous feature kernels performs better than any individual feature kernel.

Discriminative classifiers have been successfully applied to many music information retrieval (MIR) tasks. Mandel and Ellis Mandel and Ellis (2005) train SVMs on patterns in the mean and co-variances of a song's MFCC features to detect artists. Meng and Shawe-Taylor Meng et al. (2005) use a multivariate autoregressive model to describe songs. Using Jebara et al.'s probability product kernels Jebara et al. (2004), they

kernelize the information contained in these generative models and then use an SVM to classify songs into 11 genres. Eck et al. use a set of boosted classifiers to map audio features onto a large set of social tags collected from the Web Eck et al. (2007).

5.1 Sources of Music Information

We experiment with a number of popular MIR feature sets which we describe briefly here. In particular, we consider feature sets which attempt to represent different aspects of music: timbre, harmony and social context.

5.1.1 Representing Audio Content

Audio content features are extracted directly from the audio waveform. Each audio track is represented as a set of feature vectors, $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, where each item is a feature vector \mathbf{x}_t that represents an audio segment, and T depends on the length of the song. We integrate the set of feature vectors for a song into a single representation by estimating the parameters of a probability distribution, approximated with a Gaussian mixture model (GMM), over the audio feature space.

MFCCs

Mel frequency cepstral coefficients (MFCCs) are a popular feature for a number of music information retrieval tasks (e.g., Mandel and Ellis (2005); Meng et al. (2005); Flexer et al. (2006); Eck et al. (2007)). For each 22050Hz-sampled, monaural song in the data set, we compute the first 13 MFCCs for each half-overlapping short-time (~ 23 msec) segment and append the first and second instantaneous derivatives of each MFCC. This results in about 5,000 39-dimensional MFCC+delta feature vectors per 30 seconds of audio content. We summarize an entire song by modeling the distribution of its MFCC+delta features with an 8-component GMM. We consider a *global* model that samples 5,000 MFCC+delta feature vectors from random times throughout the song.

Chroma

Chroma features Goto (2006b) attempt to represent the harmonic content of a short-time window of audio by computing the spectral energy present at frequencies that correspond to each of the 12 notes in a standard chromatic scale. We extract a 12-dimensional chroma feature every $\frac{1}{4}$ second and, as with the MFCCs above, model the distribution of a song’s chroma features with a GMM.

5.1.2 Representing Social Context

We can also summarize each song in our dataset with a *annotation vector* over a vocabulary of tags. Each real-valued element of this vector indicates the relative strength of association between the song and a tag. We propose two methods for collecting this semantic information: social tags and web-mined tags. Note that the annotation vectors are, in general, *sparse* as most songs are annotated with only a few tags. A missing song-tag pair can arise for two reasons: either the tag is not relevant or the tag is relevant but nobody has annotated the song with it. In addition to being sparse, the annotation vectors tend to be *noisy* in that they do not always accurately reflect the semantic relationships between songs and tags.

Social Tags

For each song in our dataset, we attempt to collect two lists of social (raw-text) tags from the Last.fm website (www.last.fm). The first list relates the *song* to a set of tags where each tag has a *social tag score* that ranges from 0 (low) to 100 (high) as a function of both the number and diversity of users who have annotated that song with the tag. The second list associates the *artist* with tags and aggregates the tag scores for all the songs by that artist. We find all the scores for relevant song and artist tags as well as their synonyms. For example, a song is considered to be annotated with ‘down tempo’ if it has instead been annotated with ‘slow beat’. We also allow wildcard matches for tags so that, for example, ‘blues’ matches with the tags ‘delta electric blues’, ‘blues blues

blues’, and ‘rhythm & blues’. To create the LastFM annotation vector, we add the song and artist tag scores into a single vector.

Web-Mined Tags

We extract tags from a corpus of web documents using the relevance scoring (RS) algorithm, recently proposed by Knees et. al. Knees et al. (2008). To generate tags for a set of songs, the RS works by first repeatedly querying a search engine with each song title, artist name and album title to obtain a large corpus of relevant web-documents. We restrict the search to a set of musically-relevant sites. From these queries, we retain the (many-to-many) mappings between the songs and the documents. Then we use each tag as a query string to find all the relevant documents from our corpus, each with an associated relevance weight. By summing the relevance weights for the documents associated with a song, we can calculate a *web relevance score* for each song-tag pair. The song-tag scores for all tags in our vocabulary define the semantic annotation vector for a song.

5.2 Combining Multiple Source of Music Information

Given a query tag t , our goal is to find a single rank ordering of songs based on their relevance to tag t . We present three algorithms that combine the multiple sources of music information to produce such a ranking.

Both calibrated score averaging (CSA) and RankBoost directly combine the individual rank orderings provided by each of our data sources. For the social context features, these two rank orderings are constructed from the social tag score or web relevance score. For each of the two audio content features, we use the autotagging algorithm presented in Chapter 2 rank order songs. This algorithm involves learning one tag-level GMM for each tag in the vocabulary. Each tag-level GMM is trained by combining a set of relevant song-level GMMs, where each song-level GMM represents a

song associated with the tag. We then annotate (i.e., autotag) a test set song by calculating the likelihood of the song’s audio features under each of the tag-level GMMs. This produces a vector of likelihoods that, when normalized, can be interpreted as multinomial distribution over the tag vocabulary. We can then rank order the test set songs by the value of the t^{th} dimension of their respective multinomial distributions.

For kernel combination (KC), we first construct kernels for each of our data sources. For the two audio content features, we again represent songs as GMMs and compute the probability product kernel from the parameters of these distributions Jebara et al. (2004). For each of the social context features, we compute a radial basis function (RBF) kernel with entries:

$$k(a, b) = \exp\left(-\frac{\|\mathbf{x}_a - \mathbf{x}_b\|^2}{2\sigma^2}\right),$$

where $k(a, b)$ represents the similarity between \mathbf{x}_a and \mathbf{x}_b , the annotation vectors for songs a and b , as described in Section 5.1.2. The parameter σ is a hyper-parameter that needs to be learned by cross validation. If any dimension of an annotation vector (i.e., a song-tag pair) is missing, we set that dimension of the vector to zero. If a song has not been annotated with any tags, we assign that song to have the average annotation vector (i.e., the estimated vector of prior probabilities for the tags in the vocabulary).

5.2.1 Calibrated Score Averaging

Each data source produces a score $s_t(x)$ indicating how relevant tag t is for each song x in our data set. Using training data, we can learn a function $g(\cdot)$ that *calibrates* scores such that $g(s_t(x)) \approx P(t|s_t(x))$. This allows us to directly compare data sources in terms of calibrated posterior probabilities rather than incomparable scores. We use isotonic regression Zadrozny and Elkan (2002) to estimate the function g for each data source.

Recall that the social context data sources are sparse: many song-tag scores are missing. This may mean that the tag is actually relevant to the song but that no data was

found to connect them (e.g., no humans bothered to annotate the song with the tag) and we could estimate $P(t|s_t(x) = \emptyset)$ with the prior probability $P(t)$. However, we have found empirically that a missing song-tag score often suggests that the tag is truly not relevant and so we use the training data to estimate:

$$P(t|s_t(x) = \emptyset) = \frac{\#(\text{relevant songs where } s_t(x) = \emptyset)}{\#(\text{songs where } s_t(x) = \emptyset)}.$$

Once we have learned a calibration function for each data source, we convert the vector of scores for a test set song to an approximated vector of posterior probabilities. We could combine these posterior probabilities by using the arithmetic average, geometric average, median, minimum, maximum, etc. Kittler et al. (1998) In addition, we could learn a linear combination of these posterior probabilities from a validation set using linear or logistic regression. In practice, we find that the arithmetic average produces the best empirical tag-based retrieval results.

5.2.2 RankBoost

In a framework that is conceptually similar to the Adaboost algorithm, the RankBoost algorithm produces a *strong* ranking function H that is a weighted combination of *weak* ranking functions h_t Freund et al. (2003). Each weak ranking function is defined by a data source, a threshold, and a default value for missing data. For a given song, the weak ranking function is an indicator function that outputs 1 if the score for the associated data source is greater than the threshold or if the score is missing and the default value is set to 1. Otherwise, it outputs 0. During training, RankBoost iteratively builds an ensemble of weak learners and associated weights. At each iteration, the algorithm selects the weak learner (and associated weight) that maximally reduces the *rank loss* of a training data set given the current ensemble. We use the implementation of RankBoost shown in Figures 2 and 3 of Freund et al. (2003).¹

¹We also enforce the positive cumulative weight constraint for the RankBoost algorithm as suggested at the end of Section 4 in Freund et al. (2003).

5.2.3 Kernel Combination SVM

In contrast to the two previous methods of directly combining the outputs of each individual system, we could combine the sources at the feature level and produce a single ranking. Lanckriet et al. (2004) propose a linear combination of m different kernels that each encode different features of the data:

$$\mathbf{K} = \sum_i \mu_i \mathbf{K}_i, \quad \mu_i > 0 \text{ and } \mathbf{K}_i \succeq 0 \quad \forall i \quad \Rightarrow \quad \mathbf{K} \succeq 0.$$

where \mathbf{K}_i are the individual kernels formulated via the various feature extraction methods described in Section 5.1 and normalized by projection onto the unit sphere. Since each kernel \mathbf{K}_i is positive semi-definite, their positively-weighted sum is also a valid, positive semi-definite kernel Shawe-Taylor and Cristianini (2004).

The kernel combination problem reduces to learning the set of weights, μ , that combine the feature kernels, \mathbf{K}_i , into the ‘‘optimum’’ kernel, while also solving the standard SVM optimization. The optimum value of the dual problem for the single-kernel SVM is inversely proportional to the margin and is convex in the kernel, \mathbf{K} . Thus, the optimum \mathbf{K} can be learned by minimizing the function that optimizes the dual (thereby maximizing the margin) with respect to the kernel weights, μ .

$$\min_{\mu} \left\{ \max_{0 \leq \alpha \leq C, \alpha^T \mathbf{y} = 0} 2\alpha^T \mathbf{e} - \alpha^T \text{diag}(\mathbf{y}) \mathbf{K} \text{diag}(\mathbf{y}) \alpha \right\}$$

subject to: $\mu^T \mathbf{e} = 1$

$\mu_i \geq 0 \quad \forall i = 1, \dots, m,$

where now $\mathbf{K} = \sum_{i=1}^m \mu_i \mathbf{K}_i$ and \mathbf{e} is an n -vector of ones such that $\mu^T \mathbf{e} = 1$ constrains the weights μ to sum to one. This can be formalized as a quadratically-constrained quadratic program Lanckriet et al. (2004) and the solution returns a linear decision function that can be used to classify new points.

5.3 Semantic Music Retrieval Experiments

In this section, we explore the usefulness of the three algorithms presented in the previous section for the task of semantic (i.e., tag-based) music retrieval. We experiment on the CAL-500 data set: 500 songs by 500 unique artists each annotated by a minimum of 3 individuals using a 174-tag vocabulary. A song is considered to be annotated with a tag if 80% of the human annotators agree that the tag is relevant. For the experiments reported here, we consider a subset of 95 tags by requiring that each tag be associated with at least 20 songs and removing some tags that we deemed to be redundant or subjective². These tags represent genres, instruments, vocal characteristics, song usages, and other musical characteristics. We consider the CAL-500 data to be a reasonable ground truth since it is complete and redundant (i.e., multiple individuals evaluated the relevance of each tag for each song),

Given a tag (e.g., 'jazz'), the goal is to rank all songs by their relevance to the query tag (e.g. jazz songs at the top). For each of our songs, we can directly rank songs using the scores associated with the correct dimension of the annotation vector (e.g., the dimension corresponding to jazz) for that data source. Alternatively, using the SVM framework, we can learn a decision boundary for each tag (e.g., a boundary between jazz / not jazz). We then rank all test songs by their distance (positive or negative) from the decision boundary. The songs which most strongly embody the query tag should have a large positive distance from the boundary. Conversely, less semantically relevant songs should have a small or negative distance from the boundary. Reformulations of the single-kernel SVM exist which optimize for ranking Joachims (2002) but the distance from the boundary provides a monotonic ranking of the entire test set which is suitable for this semantic retrieval task.

We compare the direct and SVM ranking results to the human-annotated labels provided in the CAL-500 dataset. We evaluate the rankings using two metrics: the area under the receiver operating characteristic (ROC) curve and the mean average precision

²A complete list of the tags used will be published as supplementary material

(MAP). The ROC compares the rate of correct detections to false alarms at each point in the ranking. A perfect ranking (i.e., all the relevant songs at the top) results in an ROC area equal to one. Ranking songs randomly, we expect the ROC area to be 0.5. Mean average precision (MAP) is found by moving down the ranked list of test songs and averaging the precisions at every point where we correctly identify a new song.

One benefit of using ROC area as a metric to evaluate rankings is that it is immune to differences in the tags’ prior probabilities. The tag frequencies in the data set roughly follow an exponential distribution with most terms having far more negative than positive examples. The average prior probability over all tags is 19.6%. While a classification framework would have to overcome a bias towards the negative class, ranking songs and evaluating performance using ROC area or MAP does not suffer from this imbalance. For example, 339 of the 500 songs were annotated as having ‘Male Lead Vocals’ while only 20 songs were judged to have ‘Rapping’ vocals. In the latter case, a classifier could achieve 96% accuracy by never labeling any songs as ‘Rapping’ while its average ROC area would be 0.5 (random).

5.3.1 Single Data Source Results

For each data source (MFCC, Chroma, Social Tags, Web-mined Tags), we evaluate the direct ranking and the single-kernel SVM ranking. For SVM ranking, we construct a kernel and use it to train a one-vs-all SVM classifier for each tag where the negative examples are all songs not labeled with that tag. We train SVMs using 400 songs, find the optimum regularization parameter, C , using a validation set of 50 songs and use this final model to report results on a test set of 50 songs. The performance of each kernel, averaged using 10-fold cross validation for each tag (such that each song appears in the test set exactly once), and then averaged over the set of 95 tags, is shown on the right of Table 5.1. To be consistent with SVM ranking, we use 10-fold cross validation for direct ranking and average evaluation metrics over each fold, and then over each tag. These results appear on the left of Table 5.1.

Table 5.1: Evaluation of semantic music retrieval. All reported ROC areas and MAP values are averages over a vocabulary of 95 tags, each of which has been averaged over 10-fold cross validation. The top four rows represent the individual data source performance. “Single Source Oracle” picks the best single source for retrieval given a tag, based on the test set performance. The final three approaches combine information from the four data sources using algorithms that are described in Section 5.2. Note the performance differences between single source and multiple source algorithms are significant (one-tailed, paired t-test over the vocabulary with $\alpha = 0.05$). However, the differences between between SSO, CSA, RB and KC are not statistically significant.

Data Source	Direct Ranking		SVM Ranking	
	ROC area	MAP	ROC area	MAP
MFCC	0.723	0.443	0.711	0.423
Chroma	0.525	0.268	0.588	0.316
Social Tags	0.612	0.389	0.695	0.434
Web Documents	0.623	0.417	0.686	0.425
Single Source Oracle (SSO)	0.751	0.493	0.743	0.487
Calibrated Score Average (CSA)	0.750	0.492	.	.
RankBoost (RB)	0.747	0.485	.	.
Kernel Combination (KC)	.	.	0.740	0.478

We also show the results that could be achieved if the single best data source for each tag were known in advance and used to rank the songs. For example, the best data source for “jazz” is web documents and the best data source for “hip hop” is MFCC. This “single source oracle” can be considered an empirical upper bound since it selects the best data source for each tag based on test set performance and should be the minimum target for our combination algorithms.

Note that all four data sources produce rankings that are significantly better than random, and that MFCC produces the significantly best individual rankings³. We found that MFCC is the single best data source for about 60% of the tags while the social

³Unless otherwise noted, all statistical hypothesis tests between two algorithms are one-tailed, paired t-test over the vocabulary (sample size = 95) with $\alpha = 0.05$

context-based features are best for the other 40%. Chroma, which has the worst overall performance, is the best source for one of the tags when we use the SVM to rank the songs. This suggests that all four data sources provide useful, complimentary information.

5.3.2 Multiple Data Source Results

Using the three algorithms described in Section 5.2, we can combine information from the four data sources to significantly enhance our tag-based music retrieval system. These results are shown in the bottom three rows of Table 5.1. The best performance is achieved by CSA, though the performance is neither significantly better than RankBoost nor Kernel Combination. Note that CSA is also not significantly better than the Single Source Oracle. However, if we examine the 8 single and 3 multiple data source algorithms when considering each tag individually, 72 of the 95 tags are improved by one of the multiple data source algorithms. Specifically, CSA performs best for 15 tags, RankBoost performs best for 25, and Kernel Combination performs best for 32 tags. This suggests that each algorithm is individually useful and that combination of their outputs could further enhance semantic music retrieval.

5.4 Acknowledgments

Chapter 5, in full, is a reprint of an unpublished Computer Audition Laboratory technical report, Turnbull, Douglas; Barrington, Luke; Yazdani, Mehrdad; Lanckriet, Gert, June 2008. The dissertation author was the primary investigator and author of this papers with the exception of Section 5.2.3.

Chapter 6

Concluding Remarks and Future Directions

6.1 Concluding Remarks

The dynamics of the music industry are changing to meet the needs of music producers (musicians) and music consumers (fans) in this socio-digital age. The future role of big record companies and local record stores is in question as social networks and music download sites enter the music distribution landscape. But we do know for sure that the scale of the industry is growing at a rapid rate: 5 million artist pages on Myspace, 150 million distinct songs in the Last.fm database, 50 million iTunes customers, and 140 million iPods. As a result, new business models are emerging from both large corporate entities (Apple, Universal, Ticketmaster) and small innovative start-ups (Echo Nest, One Llama, Music Search Incorporated).

One specific area that is ripe for technical innovation is music search and discovery: *technologies that connect millions of people with millions of songs*. In this dissertation, we presented the framework for one such technology, called a *semantic music discovery engine*, that provides a natural and familiar query-by-description paradigm for retrieving music. We consider this paradigm to be *natural* because average music

consumers can use a variety of common concepts (e.g., genres, instruments, emotions, known artists and songs, etc.) to find music. We consider this to be a *familiar* paradigm because it is akin to searching for webpages using Internet search engines (e.g., Google, Yahoo, Alta Vista).

The core of the music discovery engine is a *music information index* that consists of both human annotations and automatically extracted information. The human annotations are collected in a variety of ways, including conducting surveys (CAL500), deploying annotation games (Listen Game), and collecting music reviews (site-specific webcrawling). These annotations reflect the acoustic experience one has when listening to the audio track and places the music into a social context. The major limitation of human annotations is that, while popular songs (in the long-tail) may be richly annotated with useful semantic information, less popular songs (in the short-tail) are poorly annotated or not annotated at all. To lessen the impact of this cold start problem, we developed an autotagging system that extracts semantic information directly from the audio track. This provides us with a useful automatic annotation for every song in our database.

6.2 Future Directions

Our future work can be separated into two overlapping pursuits: academic exploration and commercial development.

6.2.1 Academic Exploration

The research in this dissertation can be described as *interdisciplinary* because it involves computer audition, digital signal processing, machine learning, information retrieval, human computer interaction, user interface design, text mining, natural language processing, cognitive psychoacoustics, music theory, and musicology. As such, there a number of potential research directions for this work. In this section, we focus technical

directions that we hope to address in the near future.

Exploring music similarity with semantics

In an informal study of music recommendation systems, Lamere and Celma found that an academic system based on semantic similarity outperformed 12 commercial and academic systems based on social similarity (i.e., collaborative filtering) and five independent human experts [Lamere and Celma (2007)]. This provides a strong justification for future work involving semantic similarity. Some of our initial work focused on finding music tags that are highly predicative of music similarity. That is, if we can identify a small set of informative music tags, then we can focus on both manually collecting these tags (e.g., using a game) and automatically generating these tags from the audio content (i.e., autotagging). More importantly, we wish to further explore *similarity with semantics* so that a user can specify a heterogeneous query consisting of a seed song or artist and music tags. This will allow us to weigh specific music tags differently when calculating semantic similarity. Finally, when we present list of similar songs for the given seed song or artist, we can describe the reasons is each song was selected. That is, we produce *interpretable* results which is not the case if we use a system based on social similarity.

Combining data sources

In Chapter 4, we presented a comparison of five tag collection and generation approaches. The next step will be to focus on combining these and other sources of data using generative graphical models (e.g., three-aspect model [Yoshii et al. (2008)]), discriminative kernel method (e.g., kernel combination [Appendix ??, Lanckriet et al. (2004)], and rank aggregation (e.g., RankBoost [Freund et al. (2003)]).

Alternative Autotagging Models

The supervised multiclass labeling (SML) model presented in Chapter 2 is one of many models that have been proposed for the semantic labeling of multimedia data. Three other classes of models are supervised one-verses-all, unsupervised, and non-parametric models. (See the work of Carneiro et al. for a recent summary of models that have been developed for image annotation [Carneiro et al. (2007)].) In the context of music, Eck et al. recently proposed a supervised one-verse-all model Eck et al. (2007) and Sordo et al. proposed a non-parametric model [Sordo et al. (2007)]. To our knowledge, unsupervised learning models have not been examined for music autotagging and retrieval. In general, these models introduce a set of latent variables that encode a set of hidden states. Each state represents a joint distribution between tags and multimedia (i.e., music) features. During training, a heterogeneous data set of tags and multimedia documents (i.e., songs) is presented to an unsupervised learning algorithm, such as variational expectation maximization, or a sampling algorithm, such as a Gibbs sampler, in order to estimate the joint distribution between multimedia features and tags. During annotation, the predicted tags for an unlabeled multimedia document are the individual tags that maximize this joint distribution over all latent states. Future research will involve modifying popular models, such as Blei and Jordan's correspondence latent Dirichlet allocation (corrLDA) model [Blei and Jordan (2003)] and Feng, Manmantha and Lavrenko's multiple Bernoulli relevance (MNBR) model [Feng et al. (2004)]. Both models were originally developed for image annotation and may be adapted to take advantage of the time dependent nature of sound or correlations between music tags.

Model individuals rather than populations

While observing test subjects as they played Listen Game or participated in the CAL500 survey, we noticed that there was low inter-subject agreement for many tags. This reflects a common sentiment within the music information retrieval community: the inherent subjectivity associated with music prevents MIR systems from achieving

good performance [McKay and Fujinaga (2006)]. One way to address this problem is to focus on modeling individuals (or at least demographically or psychographically similar groups of individuals). To illustrate this point, during data collection of the CAL500 data set, we had one test subject annotate 200 of the 500 songs in our data set. A preliminary study showed that we were better able to predict some words (especially ‘usage’ words) for this subject using the 200-song subset when compared against models trained using the entire CAL500 data set. This is not surprising since we would expect an individual to be *self-consistent* when annotating songs with subjective concepts. We expect that *user-specific* models will offer us a chance to reduce the impact caused by subjectivity in music so that we can better model an individual’s notions of audio semantics.

6.2.2 Commercial Development

Music, as a form of entertainment, is a multi-billion dollar industry. We can identify a number of natural markets within this industry where our music discovery engine can have an impact:

Digital Music Market

The most obvious commercial setting for the music discovery engine is the rapidly growing digital music market (downloads and subscriptions). The U.S. market has shown rapid growth and was estimated at \$1.3 billion in 2007 (32% growth over 2006) [Card et al. (2007)]. Analysts predict this market will grow to somewhere between \$3.4 and \$4.8 billion U.S. dollars per year by 2012 when it will begin to out sell the physical CD sales. With over a dozen online music retailers boasting catalogs of over one million songs each.¹ A powerful new query-by-description interface will not only offer an intuitive way to find music within these large catalogs, but may also

¹Sites with catalogs containing over one million songs: 7digital (3.5M), Amazon MP3 (2M), Amie Street, AudioLunchbox (2M), BuyMusic, eMusic (2M), Apple iTunes (6M), MusicGiants, Napster (3M), PayPlay.fm, Puretracks, Rhapsody (4M), SpiralFrog, Walmart Music, Yahoo Music, Zune Marketplace (3M).

provide an entertaining tool that can help companies differentiate themselves from the rest of the pack.

Commercial Music Licensing

eMarketer (2007) Commercial Music Licensing, also known as synchronization, involves the licensing of music for film, television, video games, advertising, and commercial performance (e.g., retail stores, restaurants, gyms). This market was valued at \$2.1 billion U.S. dollars in 2006 and is projected to grow 4.8% per year [eMarketer (2007)]. Music distribution companies like Muzak LLC and Fluid Music are responsible for providing specialized music playlists to specific businesses. For example, JCPenney prefers passive soft rock for their conservative midwestern shoppers whereas Spencer Gifts needs edgy indie rock to attract their young and anarchic clientele. Currently, distribution companies rely on human editors to put together custom playlists to match the requirements of each business (e.g., marketing goals, cost, demographic information). Our music discovery engine is an automatic alternative that can be employed to satisfy these semantic-based requirements.

Internet Radio

Like commercial music licensing, personalized Internet radio is another human-intensive industry that could be automated using our music discovery engine. In 2006, it was estimated that there were 91 million Internet Radio users per month in the United States. This number is expected to grow to 226 million in 2020 when it will reach near parity with terrestrial radio (e.g., AM/FM). The latest development in Internet Radio is personalized streams of music based on social or semantic similarity. Pandora has been the leader in this emerging field due to the quality of their recommendations and simplicity of their web-based music player. However, despite their ability to attract customers, they are not profitable due to the high cost of their slow and laborious human annotation project. Fans also complain that their limited set of annotated songs is re-

flected by their tiresome playlists full of redundant tracks. Both problems (i.e., high cost and small corpus) can be solved using our music discovery engine.

Casual Gaming Market

Casual computer games are simple, quick, and intended for a broad audience. Some of the more successful early games include Tetris, Solitaire and Chess. More recently social casual games, like Boggle and Poker, have emerged on social networks and game portal sites. Like music, this industry is growing at a rapid rate from \$600 million U.S. dollars in 2004 to \$2 billion U.S. dollars in 2008 [Wallace and Robbins (2006)]. Based on our initial experience with Listen Game, we have observed that music provides an ideal setting for social games since people love to share, discuss and debate music. That is, even if we ignore the data that is collected by our music annotation game, the game may have (viral) potential as an entertaining platform to market music. In addition, by having users utilize semantics to describe music in the game, query-by-description may become a more familiar paradigm for finding new music.

Social Network

Many individuals, especially teenagers and young adults, use music to define themselves and to quickly assess their compatibility with others.² As a result, most social network users publicly share their preferences and many (39%) embed music directly into their profile pages [Enser (2007)]. In addition, music-oriented social networks, such as Last.fm and iLike, explicitly use music as a way to connect people to one another. Our semantic music discovery engine could be used to add a semantic dimension to this social recommendation. That is, instead of just suggesting that “Alice and Bob should be friends because they like the same music,” we can say “Alice and Bob should be friends because they both like sappy show tunes and romantic jazz music.” In addition social networks, have often been criticized for not having a clear business

²A recent study revealed that music was the single most common ice-breaker between college students [Rentfrow and Gosling (2006)]

model because virtual communities are generally wary and/or untrusting of advertisers. The ability to promote music using improved search and discovery technologies offers record labels an enhanced opportunity to place their products (e.g., music, concert tickets, merchandise).

Social-Semantic Music Portal

To summarize, both our music annotation game and our music discovery engine have broad commercial potential. As such, we envision a social-semantic music portal where users can search for, discover, listen to, interact with and socialize around music. To this end, we founded a company called Music Search Inc. This company is in its infancy, but to date, we have filed one patent, won a commercialization grant, and have had many promising connections with industry leaders and venture capitalists. In doing marketing research, we have identified Last.fm, Echo Nest, and One Llama as representing three of the more academically-inspired commercial endeavors that are moving in a similar direction.

Appendix A

Definition of Terms

In this appendix, we present a list of common terms that are use throughout this dissertation. Unless otherwise stated, our definitions are placed into a music information retrieval context.

cold start problem	the inability to retrieve a song because it has not been annotated. In general, the cold start problem affects songs or artists that are new or in the long tail.
long tail	less well known songs that are poorly annotated or not annotated at all. The long tail metaphor refers to a plot of the rank of a song (according to popularity) vs. the popularity of the song. This curve tends to be shaped like a power law probability distribution (i.e., $y = 1/x$) where most of the mass is attributed to a small number of popular songs (short-tail) and the remaining mass is distributed over a large number of unpopular songs (long tail) [Anderson (2006); Lamere and Celma (2007)].
metadata	factual information about music. See Section 1.2.

music discovery	used to find new or unexpected music based on some general criteria. The criteria may include semantic information (e.g., tags), song or artist similarity, popularity information (e.g., record charts), etc. For example, a user may want to find music that “obscure music that sounds like the Rolling Stones and is acoustic and bluesy.” See Section 1.2.
music search	finding a specific song (i.e., audio track) when the user knows the title of the song, the title of the album, or the name of the artist. For example, a friend tells you that the new Rolling Stones album is good and you want to purchase that album from a music download site (e.g., Apple iTunes). See Section 1.2.
popularity bias	more popular songs receive more attention, and as such, tend to be more richly annotated (in terms of metadata, web documents, and social tags). As a result, a more popular song will often be retrieved before a less popular song even though the less popular song may be semantically more relevant. Popularity bias is closely related to the <i>cold start problem</i> .
semantic music discovery engine	a software framework for music discovery based on a query-by-description paradigm. The term <i>discovery</i> (as opposed to <i>search</i>) is used because the system is intended to help users discover novel music, as well as uncover new connections between familiar songs and artists. The term <i>semantic</i> reflects the fact that our system is built around a <i>query-by-description</i> paradigm in which users search for music using a large, diverse set of musically-relevant concepts in a natural language setting. See Chapter 1.
short head	popular songs that are richly annotated. See long tail for a more detailed description. See Section 4.2
strongly-labeled data	data (e.g., a song) that is annotated with a tag if the tag is semantically associated with the data, and not annotated with a tag if the tags is not semantically associated with the data. See Section 4.2.

tag	a short text-based semantic token. Examples to tags (and tag categories) include “melodramatic” (adjective), “alternative rock” (genre), “conga drums” (instrument), and “sad” (emotion).
tag category	a group of semantically similar tags (e.g., genres, instruments, emotions, adjectives, usages).
tag vocabulary	a set of tags. A tag vocabulary may be structured using a hierarchy to encode relationships between tags and/or tag categories.
weakly-labeled data	data (e.g., a song) that is annotated with a tag if the tag is semantically associated with the data, but where the absence of a tag does not necessarily mean that the tag is not semantically associated with the data. See Section 4.2.

Appendix B

Related Music Discovery Projects

In addition to the core research presented in this dissertation, we have conducted four research projects that are related to the development of the semantic music search engine.

B.1 Query-by-semantic-similarity for Audio Retrieval

We improve upon query-by-example for content-based audio information retrieval by ranking items in a database based on *semantic* similarity, rather than acoustic similarity, to a query example. The retrieval system is based on semantic concept models that are learned from a training data set containing both audio examples and their text captions. Using the concept models, the audio tracks are mapped into a semantic feature space, where each dimension indicates the strength of the semantic concept. Audio retrieval is then based on ranking the database tracks by their similarity to the query in the semantic space. We experiment with both semantic- and acoustic-based retrieval systems on a sound effects database and show that the semantic-based system improves retrieval both quantitatively and qualitatively. [Barrington et al. (2007a,b)]

B.2 Tag Vocabulary Selection using Sparse Canonical Component Analysis

A musically meaningful vocabulary is one of the keystones in building a computer audition system that can model the semantics of audio content. If a word in the vocabulary is inconsistently used by human annotators, or the word is not clearly represented by the underlying acoustic representation, the word can be considered as *noisy* and should be removed from the vocabulary to denoise the modeling process. This paper proposes an approach to construct a vocabulary of predictive semantic concepts based on *sparse canonical component analysis* (sparse CCA) . Experimental results illustrate that, by identifying musically meaningful word, we can improve the performance of a previously proposed computer audition system for music annotation and retrieval. [Torres et al. (2007)]

B.3 Supervised Music Segmentation

A musical boundary is a transition between two musical segments such as a verse and a chorus. Our goal is to automatically detect musical boundaries using temporally-local audio features. We develop a set of *difference* features that indicate when there are changes in perceptual aspects (e.g., timbre, harmony, melody, rhythm) of the music. We show that many individual difference features are useful for detecting boundaries. By combining these features and formulating the problem as a supervised learning problem, we can further improve performance. This is an alternative to previous work on music segmentation which has focused on unsupervised approaches based on notions of self-similarity computed over an entire song. We evaluate performance using a publicly available data set of 100 copyright-cleared pop/rock songs, each of which has been segmented by a human expert. [Turnbull et al. (2007b)]

References

- Anderson, C., 2006: *The Long Tail: Why the Future of Business is Selling Less of More*. Hyperion.
- Aucouturier, J.-J., and Pachet, F., 2003: Representing musical genre: A state of the art. *Journal of New Music Research*, **32**(1).
- Barrington, L., Chan, A., D., and Lanckriet, G., 2007a: Audio information retrieval using semantic similarity. In *IEEE ICASSP*, II-725–II-728.
- Barrington, L., Turnbull, D., Torres, D., and G.Lanckriet, 2007b: Semantic similarity for music retrieval. *Music Information Retrieval Evaluation Exchange*.
- Barrington, L., Yazdani, M., Turnbull, D., and Lanckriet, G., 2008: Combining feature kernels for semantic music retrieval. *ISMIR*.
- Berenzweig, A., Logan, B., Ellis, D., and Whitman, B., 2004: A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 63–76.
- Blei, D. M., and Jordan, M. I., 2003: Modeling annotated data. *ACM SIGIR*, 127–134. doi:<http://doi.acm.org/10.1145/860435.860460>.
- Buchanan, C. R., 2005: *Semantic-based Audio Recognition and Retrieval*. Master's thesis, School of Informatics, University of Edinburgh.
- Buckman, J., 2006: Magnatune: free mp3 music and music licensing. [Http://www.magnatune.com](http://www.magnatune.com).
- Cannam, C., Landone, C., Sandler, M., and Bello, J. P., 2006: The sonic visualiser: A visualisation platform for semantic descriptors from musical signals. *ISMIR*.
- Cano, P., 2006: *Content-based audio Search: From Fingerprinting to Semantic Audio Retrieval*. Ph.D. thesis, University Pompeu Fabra.
- Cano, P., Batlle, E., Kalker, T., and Haitsma, J., 2005: A review of audio fingerprinting. *J. VLSI Signal Process. Syst.*, **41**(3), 271–284. ISSN 0922-5773. doi:<http://dx.doi.org/10.1007/s11265-005-4151-3>.

- Cano, P., and Koppenberger, M., 2004: Automatic sound annotation. In *IEEE workshop on Machine Learning for Signal Processing*.
- Card, D., Best, M., Guzman, A., and Mulligan, M., 2007: Us music forecast 2007 to 2012. Technical report, Jupiter Research.
- Carneiro, G., Chan, A. B., Moreno, P. J., and Vasconcelos, N., 2007: Supervised learning of semantic classes for image annotation and retrieval. *IEEE PAMI*, **29**(3), 394–410. ISSN 0162-8828. doi:<http://doi.ieeecomputersociety.org/10.1109/TPAMI.2007.61>.
- Carneiro, G., and Vasconcelos, N., 2005: Formulating semantic image annotation as a supervised learning problem. *IEEE CVPR*.
- Celma, O., Cano, P., and Herrera, P., 2006: Search sounds: An audio crawler focused on weblogs. In *ISMIR*.
- Clifford, S., 2007: Pandora's long strange trip. *Inc.com*.
- Cover, T., and Thomas, J., 1991: *Elements of Information Theory*. Wiley-Interscience.
- Dannenberg, R., Birmingham, W., and Tzanetakis, G., 2003: The musart testbed for query-by-humming evaluation.
- Dannenberg, R. B., and Hu, N., 2004: Understanding search performance in query-by-humming systems. *ISMIR*.
- Downie, J. S., 2005: Music information retrieval evaluation exchange (MIREX).
- Downie, J. S., 2007: Music information retrieval evaluation exchange MIREX wiki. [Http://www.music-ir.org/mirexwiki/index.php](http://www.music-ir.org/mirexwiki/index.php).
- Eck, D., Lamere, P., Bertin-Mahieux, T., and Green, S., 2007: Automatic generation of social tags for music recommendation. In *Neural Information Processing Systems Conference (NIPS)*.
- Eisenberg, G., Batke, J., and Sikora, T., 2004: Beatbank - an MPEG-7 compliant query by tapping system. *Audio Engineering Society Convention*.
- Ellis, D., and Poliner, G., 2007: Identifying cover songs with chroma features and dynamic programming beat tracking. *ICASSP*.
- eMarketer, 2007: Music industry licensing to drive growth. Technical report, eMarketer Inc.
- Enser, J., 2007: The digital music survey. Technical report, Oslwang.

- Essid, S., Richard, G., and David, B., 2005: Inferring efficient hierarchical taxonomies for music information retrieval tasks: Application to musical instruments. *ISMIR*.
- Feng, S. L., Manmatha, R., and Lavrenko, V., 2004: Multiple bernoulli relevance models for image and video annotation. *IEEE CVPR*.
- Flexer, A., Gouyon, F., Dixon, S., and Widmer, G., 2006: Probabilistic combination of features for music classification. *ISMIR*.
- Forsyth, D., and Fleck, M., 1997: Body plans. *IEEE CVPR*.
- Freund, Y., Iyer, R., Schapire, R., and Singer, Y., 2003: An efficient boosting algorithm for combining preferences full text pdf formatpdf (392 kb). *JMLR*, **4**, 933–969.
- Futrelle, J., and Downie, J. S., 2002: Interdisciplinary communities and research issues in music information retrieval. *ISMIR*.
- Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., , and Dahlgren, N., 1993: Darpa timit acoustic-phonetic continuous speech corpus. CD-ROM.
- Garrity, B., 2008: Ringtone sales fizzling out. *New York Post*.
- Glaser, W., Westergren, T., Stearns, J., and Kraft, J., 2006: Consumer item matching method and system. *US Patent Number 7003515*.
- Goto, M., 2003: Smartmusiciosk: music listening station with chorus-search function. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, 31–40. ACM Press, New York, NY, USA. ISBN 1-58113-636-6. doi:<http://doi.acm.org/10.1145/964696.964700>.
- Goto, M., 2004: Development of the RWC music database. In *International Congress on Acoustics*, 553–555.
- Goto, M., 2006a: AIST annotation for RWC music database. *ISMIR*.
- Goto, M., 2006b: A chorus selection detection method for musical audio singals and its application to a music listening station. *IEEE TASLP*.
- Goto, M., and Hirata, K., 2004: Recent studies on music information processing. *Acoustical Science and Technology*, **25**(4), 419–425.
- Gouyon, F., and Dixon, S., 2006: Computation rhythm description.
- Hu, X., Downie, J. S., and Ehmann, A. F., 2006: Exploiting recommended usage meta-data: Exploratory analyses. *ISMIR*, 19–22.
- Jebara, T., Kondor, R., and Howard, A., 2004: Probability product kernels. *Journal of Machine Learning Research*, **5**, 819–844. ISSN 1533-7928.

- Joachims, T., 2002: Optimizing search engines using clickthrough data. *ACM Conference on Knowledge Discovery and Data Mining*.
- Kaplan, D., 2008: Apple surpasses wal-mart as number one in u.s. music seller. *Frobes.com*.
- Kapur, A., Benning, M., and Tzanetakis, G., 2004: Query by beatboxing: Music information retrieval for the dj. *ISMIR*, 170–178.
- Kittler, J., Hatef, M., Duin, R., and Matas, J., 1998: On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **20**(3), 226–239.
- Knees, P., Pohle, T., Schedl, M., Schnitzer, D., and Seyerlehner, K., 2008: A document-centered approach to a natural language music search engine. In *ECIR*.
- Knees, P., Pohle, T., Schedl, M., and Widmer, G., 2007: A music search engine built upon audio-based and web-based similarity measures. In *ACM SIGIR*.
- Lamere, P., and Celma, O., 2007: Music recommendation tutorial notes. *ISMIR Tutorial*.
- Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M., 2004: Learning the kernel matrix with semidefinite programming. *JMLR*, **5**, 27–72.
- Law, E. L. M., von Ahn, L., and Dannenberg, R., 2007: Tagatune: a game for music and sound annotation. In *ISMIR*.
- Levy, M., and Sandler, M., 2007: A semantic space for music derived from social tags. In *ISMIR*.
- Lewis, D., 1997: Reuters-21578 text categorization test collection.
- Li, T., and Ogihara, M., 2003: Detecting emotion in music. In *ISMIR*, 239–240.
- Li, T., and Tzanetakis, G., 2003: Factors in automatic musical genre classification of audio signals. *IEEE WASPAA*.
- Mandel, M., and Ellis, D., 2005: Song-level features and support vector machines for music classification. *ISMIR*.
- Mandel, M., and Ellis, D., 2007: A web-based game for collecting music metadata. In *ISMIR*.
- McKay, C., and Fujinaga, I., 2006: Musical genre classification: Is it worth pursuing and how can it be improved? *ISMIR*.
- McKay, C., McEnnis, D., and Fujinaga, I., 2006: A large publicly accessible prototype audio database for music research. *ISMIR*, 160–163.

- McKinney, M. F., and Breebaart, J., 2003: Features for audio and music classification. *ISMIR*.
- Meng, A., Ahrendt, P., and Larsen, J., 2005: Improving music genre classification by short-time feature integration. *IEEE ICASSP*.
- Miller, F., Stiksel, M., and Jones, R., 2008: Last.fm in numbers. *Last.fm press material*.
- Nevins, C. H., 2008: eMusic keeps on growing: world's largest indie catalogue now at 3.5 million tracks. *Live-PR: Public Relations ad News*.
- Pachet, F., and Cazaly, D., 2000: A taxonomy of musical genres. *RIAO*.
- Pampalk, E., 2006: *Computational Models of Music Similarity and their Application in Music Information Retrieval*. Ph.D. thesis, Vienna University of Technology, Vienna, Austria.
- Pampalk, E., Flexer, A., and Widmer, G., 2005: Improvements of audio-based music similarity and genre classification. In *ISMIR 05*, 634–637.
- Peeters, G., 2006: Musical key estimation of audio signal based on hmm modeling of chroma vectors. *Int. Conf. on Digital Audio Effects (DAFx)*.
- Rabiner, L., and Juang, B. H., 1993: *Fundamentals of Speech Recognition*. Prentice Hall.
- Rentfrow, P. J., and Gosling, S. D., 2006: Message in a ballad: The role of music preferences in interpersonal perception. *Psychological Science*, **17**(3), 236–242. ISSN 0956-7976. doi:10.1111/j.1467-9280.2006.01691.x.
- Reynolds, D., Quatieri, T., and Dunn, R., 2000: Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, **10**, 19–41.
- Rosen, H., 2000: U.S. manufacturers' unit shipments and value chart manufacturers' unit shipments and value chart. Technical report, Recording Industry of America Association.
- Roukos, S., Graff, D., and Melamed, D., 1995: Hansard french/english. Linguistic Data Consortium.
- Sandoval, G., 2008: As expected, myspace unveils new music service. *Cnet news*.
- Shawe-Taylor, J., and Cristianini, N., 2004: *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, USA. ISBN 0521813972.
- Skowronek, J., McKinney, M., and van de Par, S., 2006: Ground-truth for automatic music mood classification. *ISMIR*, 395–396.

- Slaney, M., 2002a: Mixtures of probability experts for audio retrieval and indexing. *IEEE Multimedia and Expo*.
- Slaney, M., 2002b: Semantic-audio retrieval. *IEEE ICASSP*.
- Snider, M., 2008: Tune in, turn on to music on these five sites. *USA Today*.
- Sordo, M., Lauier, C., and Celma, O., 2007: Annotating music collections: How content-based similarity helps to propagate labels. In *ISMIR*.
- Stokes, J., 2007: Interview with dominic mazzoni of audacity. *ValueWiki*.
- Stork, D., 2000: Open data collection for training intelligent software in the open mind initiative.
- Torres, D., Turnbull, D., Barrington, L., and Lanckriet, G., 2007: Identifying words that are musically meaningful. In *ISMIR '07*, 405–410.
- Turnbull, D., Barrington, L., and Lanckriet, G., 2006: Modelling music and words using a multi-class naïve bayes approach. *ISMIR*, 254–259.
- Turnbull, D., Barrington, L., Torres, D., and Lanckriet, G., 2007a: Towards musical query-by-semantic description using the CAL500 data set. In *SIGIR '07*, 439–446.
- Turnbull, D., Barrington, L., Torres, D., and Lanckriet, G., 2008: Semantic annotation and retrieval of music and sound effects. *IEEE TASLP*, **16**(2).
- Turnbull, D., Lanckriet, G., Pampalk, E., and Goto, M., 2007b: A supervised approach for detecting boundaries in music using difference features and boosting. *International Conference on Music Information Retrieval (ISMIR)*.
- Turnbull, D., Liu, R., Barrington, L., and Lanckriet, G., 2007c: Using games to collect semantic information about music. In *ISMIR '07*.
- Typke, R., 2007: *Music Retrieval based on Melodic Similarity*. Ph.D. thesis, Utrecht University, Netherlands.
- Tzanetakis, G., and Cook, P. R., 2002: Musical genre classification of audio signals. *IEEE Transaction on Speech and Audio Processing*, **10**(5), 293–302.
- Vasconcelos, N., 2001: Image indexing with mixture hierarchies. *IEEE CVPR*, 3–10.
- Vinet, H., Herrera, P., and Pachet, F., 2002: The cuidado project. *ISMIR*.
- von Ahn, L., 2006: Games with a purpose. *IEEE Computer Magazine*, **39**(6), 92–94. ISSN 0018-9162.
- von Ahn, L., and Dabbish, L., 2004a: Labeling images with a computer game. In *ACM CHI*.

- von Ahn, L., and Dabbish, L., 2004b: Labeling images with a computer game. In *ACM CHI*.
- von Ahn, L., Ginosar, S., Kedia, M., Liu, R., and Blum, M., 2006: Improving accessibility of the web with a computer game. In *ACM CHI Notes*.
- Wallace, M., and Robbins, B., 2006: Casual games white paper. Technical report, International Game Developers Association.
- Westergren, T., 2007: Personal notes from Pandora get-together in San Diego.
- Whitman, B., 2005: *Learning the meaning of music*. Ph.D. thesis, Massachusetts Institute of Technology.
- Whitman, B., and Ellis, D., 2004: Automatic record reviews. *ISMIR*, 470–477.
- Whitman, B., and Lawrence, S., 2002: Inferring descriptions and similarity for music from community metadata. *ICMC*.
- Whitman, B., and Rifkin, R., 2002: Musical query-by-description as a multiclass learning problem. *IEEE Workshop on Multimedia Signal Processing*.
- Yoshii, K., Goto, M., Komatani, K., Ogata, T., and Okuno, H. G., 2008: An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE TASLP*.
- Zadrozny, B., and Elkan, C., 2002: Transforming classifier scores into accurate multiclass probability estimates. In *KDD*. ACM.