Learning the Meaning of Music

by

Brian A. Whitman

B.S. Computer Science, Worcester Polytechnic Institute, 1999
B.S. English Literature, Worcester Polytechnic Institute, 1999
M.S. Computer Science, Columbia University, 2001

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 2005

Author ....
Program in Media Arts and Sciences
April 29, 2005

Certified by ....
Barry L. Vercoe
Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by ....
Andrew B. Lippman
Graduate Officer, Departmental Committee on Graduate Students

Learning the Meaning of Music

by Brian A. Whitman

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on April 29, 2005,
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Abstract

Expression as complex and personal as music is not adequately represented by the signal alone. We define and model *meaning* in music as the mapping between the acoustic signal and its contextual interpretation – the 'community metadata' based on popularity, description and personal reaction, collected from reviews, usage, and discussion. In this thesis we present a framework for capturing community metadata from free text sources, audio representations general enough to work across domains of music, and a machine learning framework for learning the relationship between the music signals and the contextual reaction iteratively at a large scale.

Our work is evaluated and applied as *semantic basis functions* – meaning classifiers that are used to maximize semantic content in a perceptual signal. This process improves upon statistical methods of rank reduction as it aims to model a community's reaction to perception instead of relationships found in the signal alone. We show increased accuracy of common music retrieval tasks with audio projected through semantic basis functions. We also evaluate our models in a 'query-by-description' task for music, where we predict description and community interpretation of audio. These unbiased learning approaches show superior accuracy in music and multimedia intelligence tasks such as similarity, classification and recommendation.

Thesis supervisor: Barry L. Vercoe, D.M.A.
Title: Professor of Media Arts and Sciences

Thesis Committee

Thesis supervisor ....

Barry L. Vercoe
Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Thesis reader ....

Daniel P.W. Ellis
Assistant Professor of Electrical Engineering
Columbia University

Thesis reader ....

Deb K. Roy
Associate Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Acknowledgments

Some of this work reflects collaborations with Barry Vercoe, Deb Roy, Ben Recht, Ryan Rifkin, Youngmoo Kim and Paris Smaragdis at MIT; Beth Logan at HP Labs; Dan Ellis and Adam Berenzweig at Columbia; Steve Lawrence and Gary Flake at NEC.

Thanks: Barry Vercoe for cultivating an amazing work environment and all the advice and support over the years. All of his students, especially Youngmoo Kim for improving my phase coherence, Paris Smaragdis for the hacker and rank reductionist inspiration (and employment), Keith Martin for the music talk (and employment), Mike Casey for necessary structure (and employment), Wei Chai for four years of subliminal Chinese lessons. Rebecca Reich, Judy Brown, Nyssim Lefford, Victor Adan, John Harrison. Tamara Hearn, Kristie Thompson, Pat Solakoff and Linda Peterson.

Dan Ellis for being my academic conscience.

Deb Roy for the grounding.

Steve Lawrence, Gary Flake, Lee Giles and David Waltz for giving me my first chance to try this.

Kelly Dobson for the excitement and inspiration. Noah Vawter for years of sound ecology lessons.

Scott Katz, Tania Castaneda and Ruth Girardi, Ani Nenkova, Noemie Elhadad, Kurt Ralske, Douglas Repetto, Matt Shultz, Drew Daniel. Ben Recht for tireless explanations of math and kernel tricks. Lauren Kroiz, Jason Taylor for the extra machines, Tristan Jehan for all the coffee and tea, Mary Farbood, Beth Logan, Rob Aimi. Ryan Rifkin. Chris Csikszentmihalyi. Hugo Solis, Ryan McKinley, Gemma Shusterman, Cameron Marlow, Peter Gorniak, Hyun-Yeul Lee, Aggelos Bletsas, Ethan Bordeaux. Adam Berenzweig, Mike Mandel, David Alexander. Dan Ariely.

My father Bruce Whitman, for supporting all of this in every possible way. My brothers and musical influences Keith Fullerton and Craig John Whitman, also Sara Whitman and mules Stanley and Albert along with Robyn Belair and cat Wilbur. Special thanks to Sofie Lexington Whitman.

To the memory of my mother, Alison Lee Palfrey Fullerton Whitman.

Table of Contents

List of Figures

CHAPTER ONE
The Meaning of Music

"What's that sound coming in from the side there?"

"Which side?"

"The left."

"You mean that sound that sounds like the cutting edge of life? That sounds like polar bears crossing Arctic ice pans? That sounds like a herd of musk ox in full flight? That sounds like male walruses diving to the bottom of the sea? That sounds like fumaroles smoking on the slopes of Mount Katmai? That sounds like the wild turkey walking through the deep, soft forest? That sounds like beavers chewing trees in an Appalachian marsh? That sounds like an oyster fungus growing on an aspen trunk? That sounds like a mule deer wandering a montane of the Sierra Nevada? That sounds like prairie dogs kissing? That sounds like witch grass tumbling or a river meandering? That sounds like manatees munching seaweed at Cape Sable? That sounds like coatimundis moving in packs across the face of Arkansas? That sounds like – "

Donald Barthelme, "The King of Jazz"

1.1 Six Seconds

In late 2000, a student of accomplished avant-garde flutist James Newton expressed interest about his professor's past work with popular hip-hop / rap group the Beastie Boys. A surprised Newton, having never heard of the Beastie Boys, quickly discovered that a six second sample from his 1982 piece "Choir" was looped some forty times during the length of the song "Pass the Mic," a very successful 1992 top 40 single by the rap group. The sample was featured at the beginning of the song alone and then used as a background component to the rest of the song, overlaid with a simple drum beat and rapping by each of the Beastie Boys in turn. Enraged, Newton quickly filed suit against the group, their record label, manager, producer and media distributors for copyright violation.

What separates this case from most copyright law scuffles is that both sides maintain that the Beastie Boys followed the letter of the law: they bought the rights to the six-second segment, comprising roughly 529,200 16-bit precision floating point numbers, a megabyte of information, for "whatever use necessary in perpetuity" from Newton's record label ECM for $1,000. ECM had never told Newton of the deal, nor were they bound to, as Newton himself sold the rights to the entire sound recording where "Choir" sits to ECM for $5,000 as part of a standard recording contract. The rights for usage of the signal passed then from Newton to his record company to the Beastie Boys. If a new musician were to sample Newton's six second performance they would now have to pay the Beastie Boys' record label for the rights.

Copyright for music is normally divided into two tracks: the 'mechanical' rights which cover the sound recording and whatever means necessary to play back the recording, and the songwriting rights, which protect the ideas behind the recording: the score, lyrics and arrangements. Mechanical rights are generally purchased by the artists' record company while the artist (or composer) retains control of songwriting rights through their personal corporation (in this case, Newton's JANEW Music.) If a musician wanted to 'cover' or perform Newton's "Choir" on their own recording or on stage, they are bound to JANEW's terms. Newton's main argument in his suit rests on the fact that the Beastie Boys' corporation never contacted JANEW Music for rights to the songwriting. But sampling artists such as the Beastie Boys rarely consider songwriting rights, as their trade is in the recontextualization of the sound sample: shaping, rearranging and reforming those 529,000 numbers into their own work. Their belief is that the signal contains only the information necessary to re-create a sound. All permutations on this signal, including truncating it to six seconds, looping it and mixing it with their own work are allowed. The instant the sampling artist performs this selectional surgery all notions of musical meaning are instantly re-appropriated from the sampled artist to the artists sampling. In their view, the signal did not contain any message more important than to direct the movements in voltage a speaker should make upon receiving it. As the song created by mixing samples from various sources is now the Beastie Boys' own, why would they consider the source signals to contain their own identity? They are building blocks: tools that every musician can use to create music from "nothing." In an unpublished letter explaining their defense to the editors at the Washington Post, the rap group likens the sample they purchased to a sample of an audiobook reader saying "as well as:" a simple phrase, meaningless without context, containing no identity beyond the voice used to say it.

Obviously, Newton disagrees. The six second sample from "Choir" consists of three notes: C, D♭, C, played on a flute using a technique Newton terms "multiphonics:" singing through the flute while fingering the notes. Newton considers his multiphonics to be a unique sound, integral to his artistic identity. He states in a letter calling for support that "there is a spectrograph that moves wildly when my multiphonics are played." Newton has spent thousands in legal fees trying to receive compensation for not just the usage of 529,200 numbers which he created by playing the flute, but for the *meaning* of those numbers. When he hears the introduction to "Pass the Mic" he doesn't just hear a flute, he hears *his* flute playing, he hears the multiphonics and his style of playing, and he hears the artistic intent behind the flute playing, the "four black women singing in a church in rural Arkansas" that inspired the piece. This extra-signal information is worth just as much, and probably more, than the original signal. Newton believes that he encoded these features into the audio well enough so that his listeners could achieve the same reaction as he did upon composition. And even the smallest segment of his playing encodes this information. When the Beastie Boys sampled his recording they took far more than the signal, even if the signal was all they took. Where can we find the rest?

1.2 Music Understanding and Music Retrieval

We want to find more music that we like. We want to give something or somebody a list of our favorite groups and be given a list of music we've never heard before. We want to know who's popular this week in California, who'll be popular next week in New York. We want to have our portable device know which songs to play, only play jazz late at night, or never play anything sad when it's raining. But we're faced with a glut of data that gets worse every day and careening standards and copyright miasmas, and yet we still search for our music by filename, simple metadata such as artist or album title, or through sales-based recommendation systems. Computers are better at making sense of large amounts of data: they have more patience and don't give up so easily. The goal of our work is to make machines link music to semantic features or the outside world for the purposes of organization, recommendation, or classification. If we do it right, they'll have the same knowledge about the music as the aggregate of your entire community: they can tell you about similar sounding music, or recommend new artists no one has heard yet, or make playlists for you.

Currently the field of *music retrieval* has followed alongside text retrieval for inspiration of semantic tagging and organization techniques. Characters became samples, words became frames, documents became songs. Currently we express music as either a feature vector of signal-derived statistics, approximating the ear or a speaker as in machine listening approaches, or we express music only as its effect: marketing data from sales, shared collections, or lists of favorite songs. With the signal-only approaches we can predict with some accuracy the genre or style of a piece of music, or compute acoustic similarity, or detect what instruments are being used in which key, or discern the high-level structure of music to tease apart verse from chorus.

Some of these approaches ignore the *meaning* of music: what happens in between the music and the reaction. The retrieval systems don't know what songs are *about* or how they make people feel. They don't understand why some artists are currently selling millions of records. They are stuck inside a perceptual box: only being able to feel the vibrations without truly experiencing the effect of music or its cause. Newton's problem was that he heard his identity along with the signal, and didn't want to see it re-applied in a new context. His defendants never considered it.

In this thesis we will be learning the meaning of music. To do this, we need to first find where this extra-signal information of reaction and emotion is and find a way to represent it in the same manner that we currently represent a music signal. Then we need to ground this reaction and message in the perceptual data, entwining the context

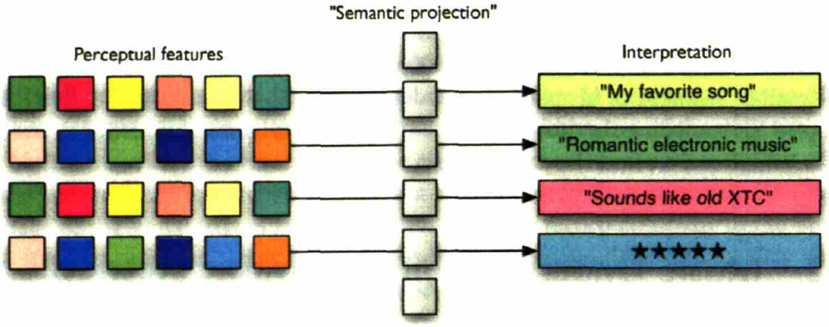


Figure 1-1: Our problem: music data through some projection leads to interpretation. The projection is an unknown process that may or may not be easy to learn.

with the content in such a way that the resultant model acts as a meaning classifier able to predict new contextual information on as-yet-unheard music. In effect we learn the 'semantic projection' that links perceptual data to the outside world (Figure 1-1) which might be derived from an entire community or from a personal user model. When we do this, we find that machines do a far better job of tying music to semantic classification, which helps computers organize, classify and recommend new music for people.

1.3 Meaning and Information in Music

It should be clear to all listeners that there is something outside the signal that either adds to or complements the larger 'musical experience.' One needs to look no farther than the matter of personal taste: you and your cousin in Tennessee might both have a strong emotional reaction to country music, but that reaction invariably widely differs. The signal coming over the airwaves, pulsing the radio's speakers and modulating your eardrum biologically is the same in both places, but somewhere in between the performer's message and your reaction is a coding scheme that takes into account some extra-signal information. If we view the flow of a musical message akin to Shannon's [69] information theory schema (Figure 1-2), a musical message is interpreted through a source (the performer) and then run through a channel which is susceptible to noise: cultural factors, buzz and trends, and other marketing peculiarities. After the channel delivers the message to the receiver (the listener) they perform their own personal decoding on the content which involves their past history with similar music, direct connections to emotions and experiences. The received message is usually strikingly different from the source message.

This whole process multiplexes a perceptible audio signal with some outside contextual information. Immediately after the musical message is formed by the 'transmitter' they are entangled: one can't exist without the other. But to date, most perceptual

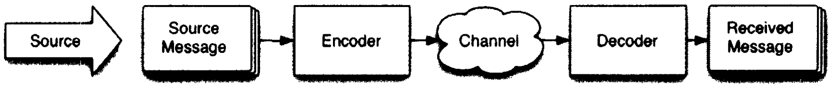


Figure 1-2: Shannon's general communication schema.

analysis of music and other media concentrates solely on the signal. Music similarity research that aims to mimic the listener's reaction to two different musical messages currently only investigates the process from the performer's point of view via signal analysis, ignoring the channel or listener. Likewise, recommendation systems that track aggregated sales data as in collaborative filtering [61] [70] ignore the personal reaction as well as anything to the left of the channel. Any musical analysis approach that ignores the contextual extra-signal information is doomed in the long run: they are making decisions based on a very small part of the musical experience.

What we attempt to do here is computationally understand this extra-signal information, and link it to the signal in such a way that it can be predicted for future audio. The benefits of such an approach are obvious: personal reaction is not easy to come by, and cultural transformations can only be studied for perception that elicits a cultural reaction. A model of the contextual information given a signal allows us to accurately 'understand' music (extract semantic features or link to the outside world) that hasn't even been heard yet. So what we call *meaning* throughout this thesis is defined as the *relationship between the signal and its interpretation.* In our work we create predictive 'machines' that analyze audio signals and extract projected community and personal reactions: these are 'meaning classifiers.'

The use of the word 'meaning' raises hackles for its vagueness; it's an overloaded term that is considerably defined by personal interpretation. We try to bridge together three distinct types of meaning into one global representation. At the outset we should make it clear that our definition of meaning above is mostly *referential*, that is, it exists as the connection between two representations. This contrasts with the purely *absolutist* view discussed by Meyer [50], in which the meaning is encompassed purely within the composition or signal. Our approach considers both with an emphasis on referential types of meaning. Many musicologists study the absolutist view of musical meaning simply because there is no formal mechanism of analyzing the contextual information. What this thesis presents are ways of computationally representing both signal-derived and contextual music information and then ways of learning a model to link the two.

1.3.1 Three types of musical meaning

We consider three types of musical meaning that have analogs in philosophical and linguistic definitions.

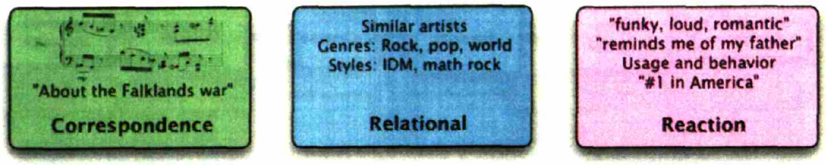


Figure 1-3: Types of musical meaning.

1: Correspondence

The composer or performer's transcription of the musical idea into a musical signal is known as *correspondence*: the relationship between the message and its representation (score, audio signal.) Any number of performers can perform an infinite amount of transformations between the idea and the signal through different interpretations. Put simply, correspondence is either "what the music is about" or "how the music was made." Sometimes the correspondence is still accessible from the signal, entangled with the audio. Newton's "four black women singing" is a form of correspondence meaning that his defendants did not immediately discover. Songs with vocals often tend to directly encode this type of meaning in the lyrics, which on their own can be studied with relative success [46]. Even with high level language analysis we are still sometimes left in the dark: in our example above in Figure 1-3 we cite the story behind Elvis Costello and Robert Wyatt's popular ballad "Shipbuilding," [19], a song with oblique lyrics whose meaning eluded us until we found a review of the song by Stewart Mason on the popular music metadata web site All Music Guide [AMG] [1]:

> The Falklands War brought a string of protest songs to the UK, most of them released well after the abbreviated conflict was over ... Set in a depressed coastal town, the song starts with the simple question "Is it worth it?," but in a flash, it moves from the abstract to the intimately personal, as the song's primary conflict is revealed: new contracts to build warships bring much-needed money to the community, but they also promise devastation to the very same working-class families whose sons will be fighting in them. [1]

As the song in question was released in 1983 and continues to find fans well after the close of the conflict with no explicit mention of "Falklands" or even war in the lyrics, there has to be a way to tie together the audio and the implied message to fully understand the music. Metadata sites such as AMG provide such a service for listeners as do music history books, record reviews and artist biographies but current music understanding systems ignore such information.

There are also forms of *explicit correspondence*, such as the notes and structure in a composition, the instruments used, the timbres and effects where they act as machines to generate the musical message to its recorded signal. These can be thought of as absolutist forms of meaning, although the choice of representation and model by the experimenter does link it to the outside world. (For example, a speech feature or model of the human ear connects the signal to human physiology.) Some of this somewhat absolutist meaning can be directly inferred from the signal, and we do not explicitly capture this in our work. Work in instrument identification [47] or automatic transcription [40] are examples of signal-only analyses of music that attempt to extract this sort of correspondence.

2: Relational

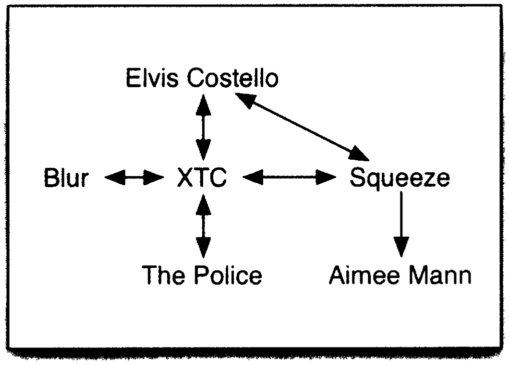


Figure 1-4: Similar artist network starting with Elvis Costello.

Relational meaning is the connection between the music and other music. In linguistic terms, the meaning of a word can be understood as its relationships (synonyms, antonym, definitional) to other words: text ontologies such as WordNet [51] take this view as does the "flat" dictionary and thesaurus to some extent. in music, understanding the connections between artists and songs lends a valuable insight into personal preference and organization along with bigger questions of music similarity.

Historically relation has been the one form of meaning captured in music organization and understanding systems– but in the starved implementation of genres. Popular music is usually split into the "big five" genres of Rock, Pop, World, Electronic and Jazz, each ostensibly a cluster of music in which all members of a group share some common intra- and extra-signal features. The basic problem with these labels is that they themselves do not *mean* anything– there is no accurate definition or correspondence between the term Rock and the music that has been labeled 'Rock.' There is often unexplainable overlap between adjacent genres, and worst of all as music changes over time, the labels have stayed the same. There is no worse offender than 'World' music, whose only possible definition is 'non-American' music. Music understanding systems that claim to predict genre in music [79] are in fact performing a type of relational meaning extraction, but the task can be thought akin to prediction of album sticker price or contract amount from audio: meaningless targets to the music experience. The notion of musical *style* is a bit better off due to their organic nature: they evolve over time and are

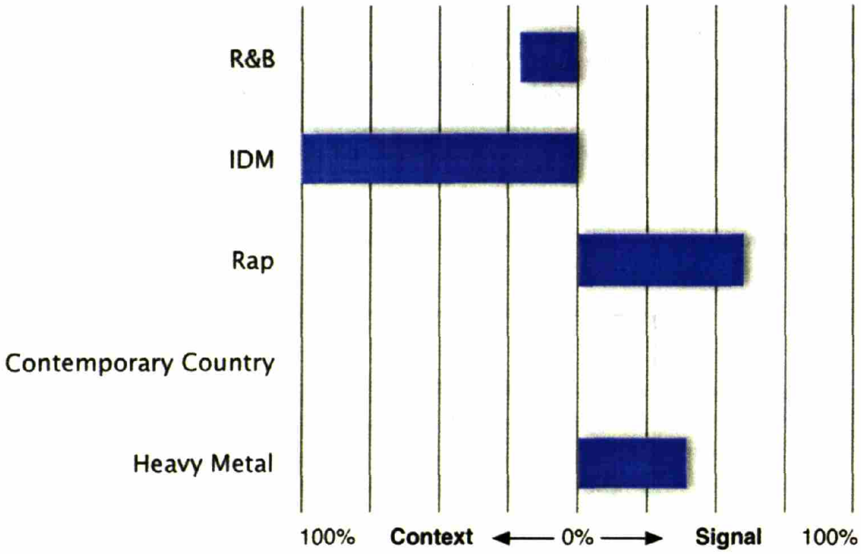


Figure 1-5: Delta between style identification results with context-based classifiers vs. signal based classifiers. [85]

not bound to sales patterns or radio charts for their success. Styles are an edited form of relational meaning that link a small subset of artists into a like-minded cluster. For example, the style 'intelligent dance music' [IDM] was conceived of in the mid-1990s to describe a set of artists that were breaking the boundaries of then-current dance music by investigating new technologies and compositional methods. It would be hard to train an acoustic classifier to predict membership in IDM (and in [85] we tried and failed) as the signal-content widely varies. But styles such as IDM have a strong *cultural* binding– invented to capture a new ideology of music making, not any particular sound– and in fact a cultural classifier trained on text patterns of description fared far better in predicting membership. In Figure 1-5 there are more results of this nature: note that contemporary country was predicted equally well by both types of classifiers, and rap and heavy metal performed better with a signal-based classifier.

We also consider the notion of "similar artists" in our work as part of relational meaning. Often a new artist will be described in terms of an already well known artist to bootstrap the experience to listeners. In our "Shipbuilding" example the songwriter Clive Langer says in an interview that the song "fuses the English charm of songs like 'When You Are A King' and McCartney's 'Junk' with the charged rhythmic restraint of Ketty Lester's 'Love Letters'" [3] and the song's lyricist and performer Elvis Costello is often said to be the "the best songwriter since Bob Dylan." [1]. Sometimes this sort of meaning is extractable from audio analysis alone, in the case where the two songs or artists are in fact sonically similar– but more often than not there are other contextual clues that relate two pieces of music or artists in some fashion. Again, edited sources such as AMG are valuable for this sort of meaning along with the formulaic sounds-like segment of most record reviews.

3: Reaction

The last form of meaning we consider important to the music experience is reaction meaning, or actionable meaning. Linguistically some claim that the meaning of a message is the action that it conducts: the meaning of "I'm angry with you" is the effect is has on the listener. Music has a similar effect in the message, it conveys various emotions (sadness, joy) or physical states (dancing) or reminds a listener of a past experience. Music also causes listeners to *listen* to it: this is best captured as personal usage data. Understanding reaction is necessary for any musically intelligent system; to ignore it removes the listener from the path of music entirely.

Significance, a form of reaction meaning, approaches the closest dictionary definition of 'meaningful' for most listeners. It is a form of aggregated *cultural reaction* that directly informs preference, notice, or popularity. Significance is captured by the cultural channel that filters a musical message towards a listener and is embedded in the buzz, trends, peer opinion, and critical reaction. We more directly consider significance to be closely related to usage or appearances in discussion in a community. Significance is a valuable form of extra-signal meaning as it can directly relate to preference and taste, along with influencing factors of future music. Current music understanding systems such as collaborative filtering recommenders do take significance into account as sale data, but often they do not grasp a larger view of trends and buzz that is not reflected in sales (as is the case for independent artists.)

It bears noting the relationship between reaction meaning and correspondence: one is attached by a community or person and one is attached by the original artist. Oftentimes the two meanings vastly differ and this is a beautiful quirk that defines the music experience.

1.3.2 Interpretation

Looking back on the three types of meaning we'll be dealing with, we should come back to our original definition of meaning as *the relationship between the signal and its interpretation.* In this work 'signal' is taken to mean the musical signal, commonly represented to a computer by discrete-time samples, but also the vibrations that eventually reach the ear. It can also be applied to analyses of the score or composition. The three types of meaning above all relate this signal to some form of interpretation: how the artist interpreted the musical idea, how the community interpreted the importance of the artist and its relationship to others, and how the listener personally interpreted the musical idea. Our meaning of music is then always found in a connection between musical content and interpretation.

We note that the meaning as stated is not always extractable as the connection between content and interpretation. Many types of meaning we catalog in this thesis are not predictable from an audio signal; we can not reliably train a system to detect if a song is about the Falklands war by the signal, nor can we predict (with any accuracy) if a song will be influential or significant, but we can access this meaning from the contextual representation of music already catalogued. Extracting meaning is not just the process of connecting signal to interpretation, it is representing the *result* of that process: indexing lyrics or reviews captures correspondence meaning without any signal analysis involved, as does analyzing sale data and radio playlists for reaction. Our work concentrates on *predictive* approaches to meaning extraction, and as such we tackle both the problems of representing meaning along with the problem of extracting meaning where none yet exists in our representation. As we are driven by the problem of music understanding and personalization to help people find music, our goal is to extract the meaning of 'unheard' (by a community) music. However, the work detailed in this thesis also proves useful for extracting meaning from contextual sources.

1.4 Our Approach

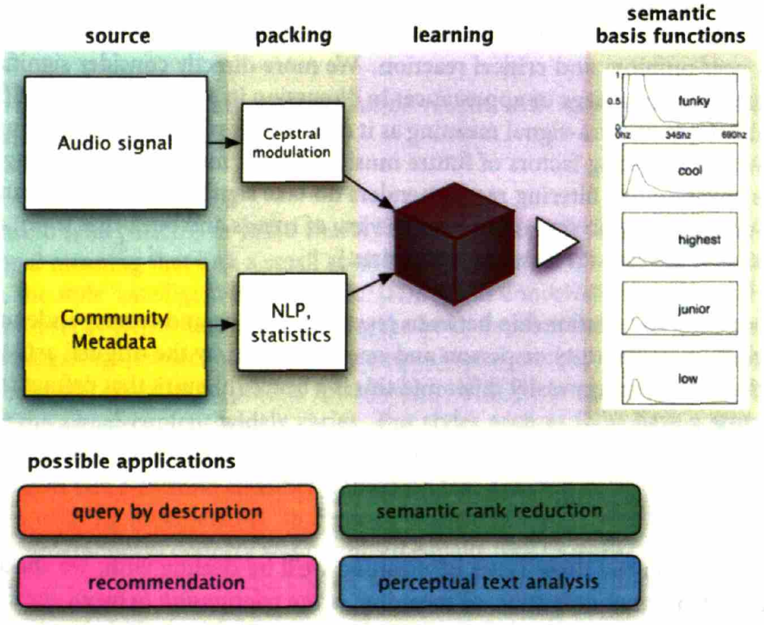


Figure 1-6: Overview of our approach.

In this thesis we attempt to learn the meaning of music by combined analysis of content and context of music along with an algorithm that learns the relationship between the two for predictive meaning extraction of new music. In our approach the contextual emphasis is on the language of music description. While the system 'listens' it also 'reads' about the music, automatically monitoring communities of listeners for their reactions to the audio as description and usage. The connection between perception and reaction is then learned with a statistical machine learning model. The outputs of this model are 'semantic basis functions,' meaning classifiers that can predict description given a new piece of audio. We show significantly improved results in common music retrieval tasks such as artist identification and music similarity using our approach, which lends validity to our thesis: that meaning extraction in music tasks is a necessary step to fully representing music in a machine.

Our approach is almost entirely unsupervised and automatic. Our audio data is randomly chosen from a large testbed and the cultural metadata is extracted from free web text and internet service crawls. At no point do the experimenters train specific models of genres, or self-select sets of artists, or choose which descriptive terms to learn – the community defines the acoustic content and the descriptive metadata. From a machine learning standpoint our process attempts to guess the target labels and ground truth from the community. This process is iterative and computationally hungry – we use a cluster of computers and sections of this work will discuss how to deal with parallelizing common signal processing, language analysis and machine learning tasks. The system can work across languages and all genres of music, and we also discuss approaches in non-music acquisition domains such as images.

As seen in Figure 1-6, we perform audio signal analysis using the modulation cepstra feature alongside extraction of "Community Metadata," which is packed using various language processing techniques. The community metadata representation is a major contribution to representing contextual information of music; many types of meaning can be directly inferred from this representation. However, together they create a predictive model (the "black box,") in our implementation using regularized least-squares classification (RLSC). We intend this work to be a robust and accurate musical understanding system as it is the first to attempt to understand both the signal and the listener's interpretation.

The output of our system include the community metadata representation as well as the predictive semantic basis functions. In this thesis we concentrate on and evaluate three specific applications using these outputs: semantic rank reduction (a way to measure the effect of integrating meaning into a music understanding system,) query by description, and perceptual text retrieval. Overall, the meaning connection between acoustic analysis and community modeling reveals a number of possible solutions:

- **Query-by-description as a music retrieval interface.** The community analysis and language modeling work can be used directly as a front end to a natural query interface to popular artists, or linked to the audio to automatically describe new music.
- **Buzz prediction, popularity analysis.** The community models of description and usage can be collated into a global prediction model of popularity and buzz. Community metadata can also be used to predict effects of new audio.
- **Perceptual text analysis and text retrieval enhancements.** With an understanding of what perceptible data terms refer to, we perform more accurate summarization and better similarity retrieval on new descriptive text (such as reviews and discussion.)

- **Semantic rank reduction.** Alternatives to statistical decomposition techniques for multimedia retrieval and analysis: de-correlating music's semantic attributes instead of its acoustical statistics for far better accuracy in signal-level multimedia intelligence tasks.

- **Trusted and natural recommendation.** A recommender that knows not to offer music from the mid-80s, or knows what you think 'sad' means might finally change the public's perception of marketing-led collaborative filtering systems as music-blind sales agents.

1.4.1 Meaning Types Considered in Our Approach

To connect our specific approach back to the meaning types we discussed in Section 1.3.1, we note that we mostly find meaning as reaction in our community metadata representation and its link to the signal due to the nature of our data mining techniques. The strongest connection between signal and interpretation comes from personal reaction to music (such as 'loud,' 'romantic,' 'angry'.) The notion of correspondence meaning is captured in the community metadata but would rarely appear in the learned predictive link. Relational meaning is captured in the community metadata through similar artists and songs, and can be predicted with good accuracy from the learned model.

Since our data collection approach for the contextual representation (covered in Chapter 4) is based on free (unstructured) text analysis, higher level notions of song stories and involved hierarchies of musical influence and imitation are only captured very broadly. Our community metadata relates 'war' to the song context "Shipbuilding" only as a probability of salience, not as a position in a knowledge base. As a result our approach works best for reaction-type meaning such as 'funky,' 'loud' and 'romantic' due to the simplicity of the terms and their high probability of being used in music discussion. That said, our approach does embody the idea of meaning as correspondence: a text-only search through our contextual representation would reveal 'about the Falklands war' for only few songs. We do note that this connection is small and not adequately dealt with in our first pass at the problem.

As the majority of our work considers the procedural link between audio perception and reaction ('grounding,') our lack of a strong representation for correspondence meaning does not get in the way of proving that meaning (or, more specifically, relational and reaction meaning) is useful for music understanding tasks. Correspondence is normally not a type of meaning that can be predicted without contextual knowledge, and for our evaluation and model of 'semantic basis functions,' reaction and relational meaning can and does show that our learning algorithms are positively affecting our task.

1.5 Layout of Thesis

We first discuss the current state of the art in music retrieval as well as the relevant research in grounding terms to perception and machine learning. We then look at various acoustic analysis techniques for music understanding. Next, we cover our work in language and aggregate usage analysis of music, collated as 'community metadata.' Next, we discuss the machine learning techniques used to learn the relationship between these two representations to capture the meaning of music. We then present results in our evaluation tasks of query-by-description and artist identification, and conclude with a note on other domains of perceptual grounding.

CHAPTER TWO
Background

Our work on meaning recognition in music is supported by recent advances in perceptual 'causal grounding'– learning the meaning of words by their use and action in the perceptual domain. In this section we'll be going over ways to represent meaning in a computational model, and then move onto grounding research in speech, video, images and sound. Afterwards we present a high level overview of signal-level and cultural music retrieval, followed by a short survey of machine learning, rank reduction and parameterization methods.

2.1 Meaning Modeling

Putnam famously states that "meaning ain't in the head." [56] His two examples regarding this statement both try to make the argument that the meaning of a word cannot be defined by the *psychological state* of the speaker. In his 'twin earth' thought experiment, where a doppelganger on a planet that has a substance called 'water' with all the same uses and connections as the water back home yet has an entirely different chemical structure, both twins' 'extension' (their inside-the-head 'meaning of water') is the same. Yet since the substance (the content) is quite different, the psychological states are pointing to two different referents and therefore they did not in fact hold the meaning.

A more musically relevant example is of the difference between the elm and the beech tree, in which we have one reference to both trees in our psychological state that can be used for either tree as we most likely don't know what the difference is (we know there is a difference, however.) The implied thesis here is that meaning is *socially constructed* for the elm vs. beech division – there is a "linguistic division of labor." We can assume there is some expert to make the characterization for us if needed, but for the time being, they both exist with the same internal psychological state. Our work follows this view closely– our meaning-as-reaction and relation in music is divided among a community of listeners rather than internally in one listener. Our meaning recognizers never settle on a single view of 'romantic' or 'popular,' but rather update their models often by polling the experts. This places meaning of music far outside the head.

Our work is informed by previous attempts to model meaning to a machine. The oft-cited case of modeling linguistic meaning in a computer is WordNet [52] [51], a lexical database or ontology of words and their relationships. WordNet makes heavy use of relational meaning, embedding the meaning of a word as their relationship (in a synset) to other terms. Moving a level up to concepts, the work in distributed commonsense modeling [71] uses free text facts submitted by users which is then parsed for content and linked together in a large semantic database. Other 'knowledge databases' such as Cyc [44] and ThoughtTreasure [49] use edited knowledge information by a team of experts. ThoughtTreasure also supports spatial reasoning– representing places and objects in ASCII 'maps' directly in the database.

Meaning has a strong biological component: our relationship with music is often informed by our biological potential: music perception work in how we respond to beat [67] or melody gives us clues to the types of music we appreciate and create. Our model directly takes advantage of the biological constraints in a perceptual feature encoding that models pitch response of the ear and time response of our perceptual system.

2.2 Grounding

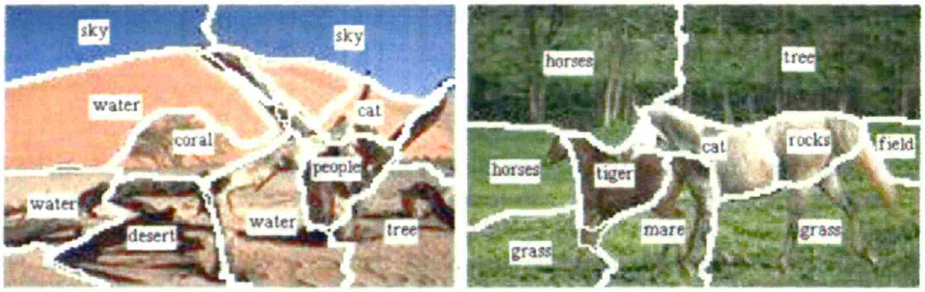


Figure 2-1: Barnard et. al's work on grounding image annotations to images.

Our links between perception and interpretation are enabled by grounded models of terms. We consider a term 'grounded' [34] if we can find a causal link between perception and reaction or interpretation – and our process of grounding then is to uncover perceptual links to language. A term on its own, such as 'loud' means little to a machine that is asked to understand it. But a term with a related perceptual representation, such as loud paired with a befitting spectrograph or other audio-derived feature or filter showing its effect, can be operated on procedurally and algorithmically. 'Loud' and 'more' grounded can beget 'louder,' 'less quiet,' 'deafening,' and an innumerable amount of other descriptively dense labels.

We note two types of grounding – causal vs. predictive. In [64] grounding is defined as the "process by which an agent relates beliefs to external physical objects." The work in our thesis is a purely *causal* form of grounding in which the perception causes a reaction. (The 'agent' or in our case machine learning apparatus has no idea of its past interactions with perception or reaction, nor the future.) A predictive grounding model approaches a formal view of meaning representation, linking the physical world not only to beliefs but to the goals and actions that can be operated on those beliefs. [64]

Work on grounding and meaning acquisition has progressed with better representations of multimedia and more robust machine learning. Grounding a large set of terms requires immense computational power to capture the degrees of freedom in description and perception, and there are often unknown constraints on the expressive power of terms. Often, we look to clues in the biological system [60]: the 'human semantic potential.' In the visual domain, some work has been undertaken attempting to learn a link between language and perception to enable a query-by-description system.

The lexicon-learning aspects in [22] study a set of fixed words applied to an image database and use a method similar to EM (expectation-maximization) to discover where in the image the terms (nouns) appear; [8] outlines similar work. This is the closest analog to our meaning recognition work in that description is learned from a large dataset of perceptual information with community-applied tags. However, in their case the image labels were applied professionally by the photo editors and using a simple grammar (usually just a list of words.) The results, however, are often worthwhile, in Figure 2-1 the object segmentation and labeling work in concert to identify the 'meaning' of the detected regions, often with good or close accuracy.

Regier has studied the visual grounding of spatial relation terms (above, across, below) across languages from a set of human experiments, ending up with a mathematical model and graphs for common prepositions [60]. Verb semantic grounding undertaken in [72] maps a logical model of action to simple video representations such as 'push' or 'pull.' In [63] color and shape terms were learned in a grounded word learning system. In the general audio domain (sound effects) recent work linked sound samples to description using the predefined labels on the sample sets [73]. The link between musical content and generalized descriptive language is not as prominent, although [20] shows that certain style-related terms such as 'lyrical' or 'frantic' can be learned from the score level. Our previous work in [84] is the first general music 'query-by-description' task, using a simple audio feature and adjective terms from community description.

2.3 Music Retrieval

Music understanding and retrieval systems have existed for some time, starting with the audio retrieval work of Jonathan Foote [29]. Retrieval tends to be either 'score level,' analyses of symbolic music data (MIDI, scores, Csound, performance data), 'audio-level,' analyses of recorded music using signal processing techniques, or recently 'cultural,' studying web communities, text, description, and usage. In our work we consider only the audio domain along with the culturally-derived extra-signal Community Metadata.

Music retrieval's goals are to provide music content understanding to the end user for categorization, classification, and recommendation. It is a timely field of research given the current marketplace of digital music distribution. It addresses a set of challenging problems:

2.3.1 Music similarity

Music similarity is concerned with the task of returning similar songs or artists given a song or set of songs in the audio or score domain. Similarity is fundamentally the back end to most other music retrieval and understanding problems, as a perfect similarity metric would indicate a "perfect music intelligence." Directly, similarity informs acoustic domain recommendation agents and playlist generators. It has both mathematical and cognitive [36] underpinnings. In [86] attempts are made to abstract the content from the style in a manner that could recognize cover versions or live versions of a song in a database. Machine learning approaches as in [12] make use of 'little experts' or anchor models: single-class models of simple music metadata (is the singer male or female, is it rock or not) pull together in similarity space to return a semantically-anchored distance measure.

In our own previous work in this field we studied a large set of music similarity judgments made by human raters. [23] We evaluated the similarity judgments against our own predictions in the audio domain (both using the above anchor models and without) and with culturally-derived data as in playlists, usage data and webtext.

2.3.2 Genre and style classification

Popularized by Tzanetakis in [79], genre ID attempts to cluster or classify audio-domain music into one of a small set of genres. Style ID [85] is the same process but at a finer grain and often requires cultural metadata. Genre ID systems claim high accuracy but are often subject to incongruence in the ground truth: few listeners can agree on the target genre of many artists (rock vs. electronic vs. pop, for example) and models vary widely depending on the music metadata source used as ground truth. Related subproblems include nationality detection in folk music [15], computed on the score level.

2.3.3 Artist Identification

First attempted in our own work [82] from low level audio features, artist ID is a challenging retrieval problem with good ground truth and still low results. A robust artist ID system could claim to understand the meaning of musical identity, as artists often change styles (and spectral 'fingerprints') but still retain complex human-identifiable features. Most attempts at artist ID currently use low-level audio features to create an evolving model of the artist's music averaged over time. In [11] the voice tracks are isolated, which improved on the task but showed problems with scale over a large set of music.

2.3.4 Music metadata extraction

In organization and library work, automatic extraction of musical features such as dominant pitch, tempo, key and structure are valuable tools. Tempo tracking's state of the art is found in [67], where a filterbank approximates the ear's response to music.

Structural analysis of audio, useful for summarization and retrieval, is approached in [18] using a similarity matrix computed over the spectral content of a song, with heuristic clustering to find the segments. Other similar approaches using different features and heuristics appear in [33] and [9]. A different approach using dynamic programming can be found in [16]. Event segmentation (extracting smaller segments than song structure components) is found in [38] for use in a time-axis redundancy reduction scheme.

2.4 Text Understanding and Retrieval

The text approaches used in this work are inspired from research in natural language processing for information retrieval. For example, in [25] extracted noun phrases are used to aid a query task. For an overview of noun phrases and their grammar, see [24]. Work in text summarization (as in [48]) attempts to cull unnecessary and repeated text from a series of documents using machine learning models of term frequencies and meanings.

Text classification work, for example to cluster documents by topic or content, is a closely related field covered in [39], where a support vector machine (SVM) learn topic classifications from thousands of 'bag of words' features. Text categorization can be applied to opinion and buzz extraction as in [21].

Text analysis can be coupled with link analysis on web pages to cluster communities for topic detection where there are no explicit topics defined as in [31] and [28].

2.5 Cultural Approaches to Music Retrieval

Collaborative filtering [70] [61] is the 'cultural' music retrieval approach that is most commonly implemented and in the widest use. In these types of approaches, users are represented as vectors containing their preferences (either sales or explicit ratings) and connections of preference are found by computing similarity metrics among users. This is a circuitous method of performing music similarity (defining similarity by its effect on users) but a valuable one that has found a strong foothold in the marketplace. Collaborative filtering approaches are bound by the 'popularity effect' or 'slow start' problems, where only well known artists or titles can be recommended– they rely on outside forces to make their similarity judgments.

Obtaining the 'cultural zeitgeist' of music perception *without* explicit ratings or sales is an important part of music understanding but only recently has there been much work relating music retrieval to text and usage mining. Our previous work [83] first defined the notion of 'Community Metadata' as applied to artist information (which will be described in greater detail in Chapter 3) and was inspired by work in [17] that extracted musical recommendations from finding lists of artists on web pages. Since then the notion of a 'cultural representation' has proven valuable for many music retrieval tasks, especially similarity [10] and style identification. [85] [53]

2.6 Machine Learning and Rank Reduction

Much work in multimedia analysis relies on a de-correlation or rank reduction step on the extracted features. Oftentimes, especially in the case of audio-derived observations, there are redundant or highly-correlated dimensions in the feature space. Supervised machine learning approaches such as learning the artist or genre of a piece can be viewed as a very low-rank semantic transform: from n dimensions of audio per frame, the system returns one. But generally researchers have computed unsupervised clustering and rank reduction transforms on their entire training dataset to then make them statistically sound for later classification or regression.

In general, removing statistical dependence of observations is used in practice to dimensionally reduce the size of datasets while retaining important perceptual features. Using tools such as Principal components analysis (PCA), researchers often reduce the dimensionality of a data set by only keeping the components of the sample vectors with large variance. By projecting onto these highly varying subspaces, the relevant statistics can be approximated by a smaller dimensional system. This provides efficiency in storage, regression, and estimation as algorithms can take advantage of the statistical compression. The main tool of Principal components analysis is the Singular Value Decomposition (SVD) [32].

Non-negative matrix factorization (NMF) [42] performs a similar decomposition as PCA but constrains its bases to be positive in an attempt to mimic part-finding in observations. We find that noisy audio observations fare better with PCA, but highly harmonic musical content (such as piano solo pieces) are a good fit for the additive nature of NMF.

Structurally-aware transforms such as Isomap [77] embed the observation space in a manifold, where the experimenter defines the distance metric that encapsulates the data. This model closely follows recent work in 'categorization by combining' [35] or the mentioned 'anchor models' [12] where a series of sub-classifier experts each feed into a larger combiner classifier. However, we note that these methods are far from unsupervised, as the experimenter must set up the semantic content in each machine, or in Isomap's case, the distance function. In effect, the bias could be embedded directly in the machinery of learning.

Our work makes extensive use of the support vector machine (SVM) [80]. The SVM is a supervised machine learning algorithm that finds the optimal separating hyperplane or regression line within a set of multi-dimensional observations. The SVM is aided by the 'kernel trick –' where data is first embedded into a Reproducing Kernel Hilbert Space (RKHS) [7] via a kernel function of the experimenter's choice. Much like Isomap, this kernel function can be thought of as an 'intelligent distance measure' but the resultant kernel must matrix satisfy the conditions of being convex and semi-definite positive (all zero or positive eigenvalues.) Much work has been done on kernel functions for various tasks, including Fourier kernels for time-aware learning [65] and geometric models [41].

CHAPTER THREE

Acoustic Analysis of Music

Machine listening [68] or computer audition [78] is concerned with extracting structure, symbols or semantic attachment from audio signals. Speech recognition using hidden markov models [57] is an application of machine listening, as is source separation [74], musical beat tracking [67], instrument identification [47] and music transcription [40]. All of these tasks first require that the algorithm view the audio data in some form that takes into account human perception, minimizes redundancy and allows for similarity and correspondence operations to be performed. In this chapter we will go over some basic methods for representing sound to a machine as a feature vector, and discuss the various design goals of a generalized music audio representation. We work towards and then describe our feature, "Penny," which is used in our meaning recognizers.

3.1 Feature Extraction for Music Understanding

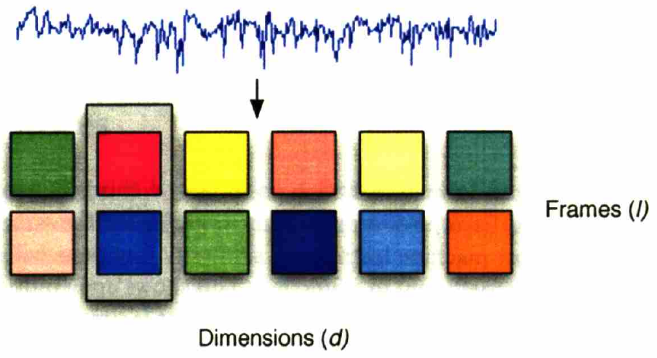


Figure 3-1: Converting a signal to a feature vector: frames (ℓ) by dimensions (d).

In our work we use a *frame-based* or discrete feature representation of musical audio. In this model, audio is decomposed into a set of ℓ vectors each with d dimension. (See

Figure 3-1 for an example.) Each frame (vector) represents a small time slice of audio, and normally the frames are aligned in chronological order. (It should be noted that most machine learning or pattern classification systems ignore the order of features.)

The features within each frame should represent some measure of the target audio, either explicitly or statistically. For example, the first dimension of the frame could indicate the overall energy of the audio frame, the second dimension could indicate a measure of harmonicity, average beat, and so on. On the other hand, each dimension could reflect the coefficient of some basis projection through the audio, or spectral content from a frequency transform. The decision is a trade-off: either an attempt is made to explicitly define 'music' by extracting and presenting the system with cleanly musical features, or the machine learning apparatus figures it out with a more bottom-up perceptual approach. The question can be stated as: "Is the intelligence in the feature or the pattern recognition?" In our work we side mostly on the machine learning, for two reasons:

- **Generalizable to all types of music:** a beat or pitch explicit feature would not work on atonal or free-form music.
- **Unknown task:** since our ground truth labels are never known until evaluation, we often do not know ahead of time what we are looking for in the music. A generalized feature space is more valuable for a task that starts with no knowledge of the target class.

There are normally a few 'ground rules' in developing a musical feature, and usually most work in conjunction with whatever learning algorithm is used. For example, if the feature vectors are treated as a matrix $\mathbf{A}$ of size $d_A \times \ell$ with a ground truth vector $\mathbf{y}$ of size $d_y \times \ell$, the 'machine' (the transformation operated on new features to predict the ground truth) is $\mathbf{x}$ as in $\mathbf{Ax} = \mathbf{y}$. Seen as a system of equations, d must always $\leq \ell$, meaning feature size should never eclipse the amount of frames in a model. There is often a relationship between ℓ and the number of classes in the target ground truth (often thought of as the dimensionality of $\mathbf{y}$, d_y) – many multi-class learning algorithms will need a significant amount of ℓ per class in $\mathbf{y}$ to properly distinguish among different classes.

Features must maintain coordination among position in d, this is shown above in Figure 3-1 as a shaded box. Each position in a frame must 'mean' the same thing among different frames, as most machine learning or similarity functions will treat each dimension as a variable or coefficient in a function. This shows up in music as a threat to time-aware feature extraction: for example, a feature that is simply the samples of audio over some short window would not work, as the position in each frame would not have any important reference to the music. However, if the song was first segmented at a beat level, and the first dimension of each frame would be guaranteed to be the samples starting with the beat, then the dimensions among different frames would be coordinated. There is usually no such restriction for adjacent dimensions: the first and second dimensions of a frame can refer to wholly separate audio features (such as dominant pitch and harmonicity), as long as they still refer to the same principles in other frames. However, most machine learning algorithms would like the dimensions to be whitened or have mean removed and set to unit variance; a simple euclidean distance between vectors that were composed of both integer note number and continuous values of energy might run into issues of normalization and scale.

The scale of the frames is one of the more important facets of feature representations in music. A single frame can refer to an entire song, a small part of a song, a whole segment, an entire artists' work, or a single discrete-time sample. Often the task should inform the scale of the feature: if performing song similarity each frame could represent an entire song; if doing artist identification some within-song scale seems to work best, and for parameterization tasks such as pitch or beat tracking a finer grained scale is necessary. The scale can be viewed as a 'control rate' in the application, determining the granularity of decision for the test or application case. For a beat tracker, for example, the decisions must be made within ± 0.1 Hz and the frame scale must reflect this.

3.1.1 Design Goals for Meaning Recognition

As mentioned above, we are looking for a generalizable music representation that makes little to no assumptions about music. We will assume that time is important for understanding music, as is the psychoacoustic response of music in humans. To this end our resultant feature space, "Penny," makes use of varying levels of structural information along with a perceptual model that scales frequencies to better approximate the ears' response.

The target semantic classification in our task varies from single-frame filters such as "loud" and "funky" but moves up to higher-level descriptors such as "driving" or "romantic." The notion of significance, correspondence and preference are also very complex targets that require a need to understand music at different structural levels. That said, since our problem is so large and must scale to thousands of artists with dozens of songs each along with tens of thousands of descriptors, we need to optimize the tradeoff between feature size and information content.

3.2 Features

In this section we'll briefly cover a few different types of features used in music understanding systems, starting with time approaches, then spectral approaches, then cepstral approaches.

3.2.1 The time domain

It's hard to derive structure by studying the signal of a test song (Costello's "Shipbuilding") as in Figure 3-2. To the trained eye one could pick out *onsets*, sample positions where a low-energy to high-energy transition is made, and perhaps the overall structure of volume could be determined. But to a frame-based machine learning system,

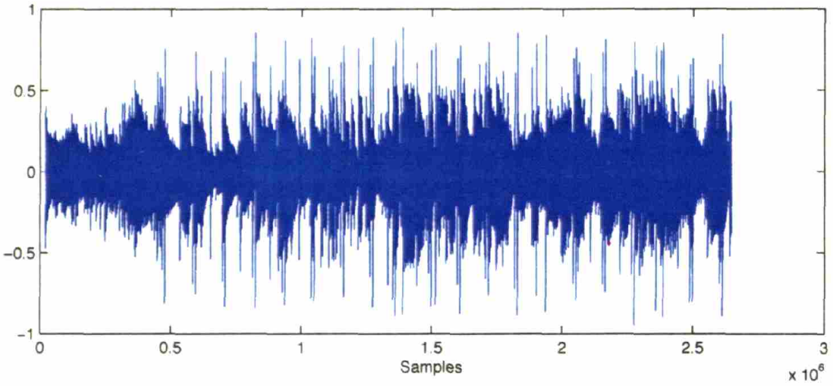


Figure 3-2: Audio signal of the first 60 seconds of "Shipbuilding."

time domain data needs to be packed into some form fit for cross-frame dimensional analysis.

Root-mean-square (RMS) Energy

One simple way of packing time domain data into frames is to take some aggregate analysis over a small window size. The spectral approaches below will tackle this in more detail, but for a simple view of the energy of the signal, one can compute the root-mean-square energy of small windows of audio. The root-mean-square energy r an audio signal frame x of length n is defined as:

$$r = \frac{\sqrt{\sum_{i=0}^{n-1} \|x\|^2}}{n} \tag{3.1}$$

To form RMS into a feature vector, we can take the RMS estimation at evenly-spaced frames of the time domain signal. For example, in our 60 second signal, approximating the RMS at 5 Hz returns 300 estimates of energy which we can align in a feature vector. Figure 3-3 shows the results of such an analysis.

3.2.2 Spectral approaches

Often we want to represent music in the frequency domain before trying to learn structure. The intuition behind this approach comes from the natural 'matrix-ness' of the short time spectral transform of a signal $X[k, j] = \mathcal{F}(x[n])$, in which given a sample window size w, a matrix of frequency × time (j frames, one for each window of length w) is generated. Since each bin $1..k$ of $X[k]$ refers to a fixed frequency in time, a machine learning system operating with the assumption that the features are coefficients will attempt find correlation among the different frames.

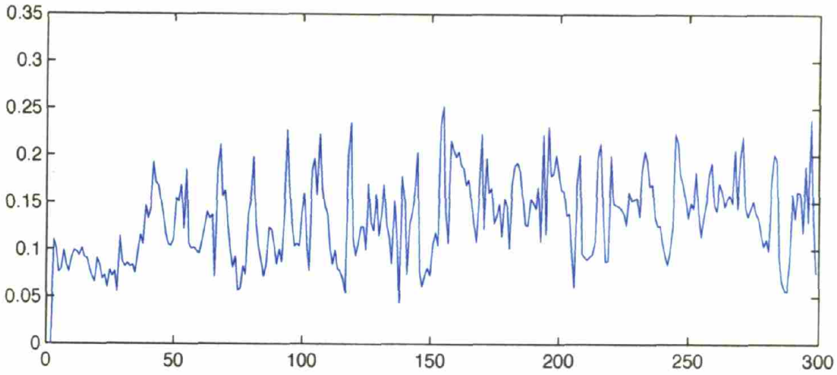


Figure 3-3: RMS energy computed in 5 Hz frames across the first 60 seconds of "Shipbuilding."

There are few machine learning algorithms that handle the instability of phase in the complex numbers of $X[k]$ as the dimensions would be inconsistent over successive frames. [59] We usually immediately take the magnitude of the spectral transformation: $X[k, j] = \|\mathcal{F}(x[n])\|$.

Power spectral density

Power spectral density (PSD) is normally interpreted as the mean of the short time spectral transform over some fixed window size. In our work we normally use a PSD window of 0.5 to 5 Hz. At the low end, for every two seconds of audio, a vector of $\frac{w}{2} + 1$ coefficients of frequency are returned. This is the 'sound of the sound:' no particular event happening in the audio will be represented, nor will time information get across if each frame is treated independently. See Figure 3-4 for the first 8 frequency bins of the first 60 seconds of "Shipbuilding" from a 5 Hz 256-point (w) PSD. Note the correlation among adjacent bands; this redundancy should be eliminated either by the learning system or a rank reduction pre-processing step.

"Beatogram" - Spectral Autocorrelation

A simple and popular permutation on the PSD or short-time Fourier transform is colloquially known as the 'beatogram,' or more directly the spectral autocorrelation. The beatogram is the "STFT of the STFT," where each vector of time variation per frequency bin is transformed with a single FFT into the frequency domain. The intuition behind the beatogram is to represent frames of music as their repetitive bases. Strong energies in the low bins of a beatogram vector represent low-frequency oscillations.

By segmenting the audio (at a rate between 0.5 and 5Hz) in the same fashion as the PSD, we take first the STFT of each sample frame, and then for each resultant row $X[i, j]$ representing spectral energy at frequency bin iF_r at frame j where F_r is the

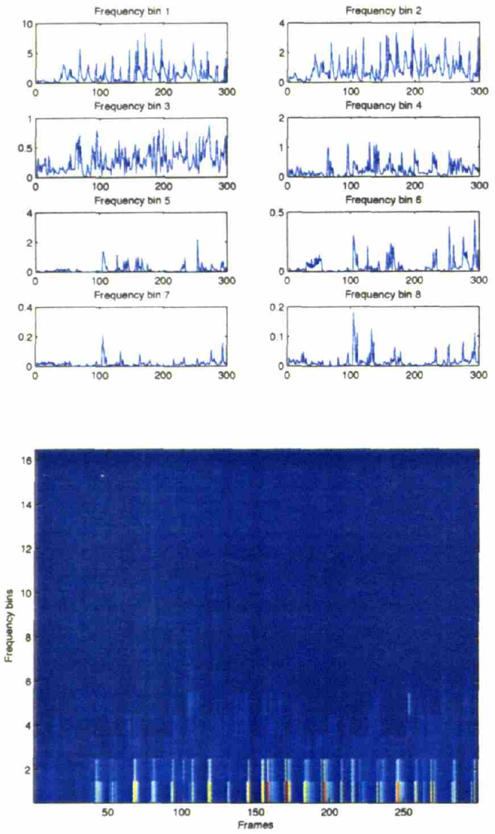


Figure 3-4: Top: power spectral density vectors for the first 8 bins of frequency information over the first 60 seconds of "Shipbuilding." Bottom: first 16 bins of frequency information arranged as a matrix.

analysis rate, we compute $\|\mathcal{F}(X[i])\|$. The resulting frequency transform represents the spectral activity's *modulation* over one period of the analysis rate. We average this modulation energy over large sections of frequency content (computing it once per $X[i]$ in $1..k$); in practice we set a cutoff between 'low frequency modulation' $(0 - \frac{F_s}{8}$, where F_s is the sampling rate) and 'high frequency modulation' $(\frac{F_s}{8} - \frac{F_s}{2}.)$ It should be noted that higher analysis rates (such as 5 Hz) do not capture musically informative modulations, we normally use a base analysis rate of 0.5 Hz for the beatogram. We then align the modulation as a column in the final beatogram, and repeat this process for each period of the base analysis. The result is a matrix $\mathbf{B}$ where $\mathbf{B}_{i,j}$ represents the modulation energy at time slice i for modulation frequency j. j ranges from DC to $\frac{F_{s\,mod}}{2}$, the base analysis rate.

In the example of "Shipbuilding" in Figure 3-5 we see that the beatogram and PSD give a different view of the same data. In this example, we show the beatogram computed on the initial frequency range of $0 - \frac{F_s}{8}$ and then the upper range of $\frac{F_s}{8} - \frac{F_s}{2}$. The intuition

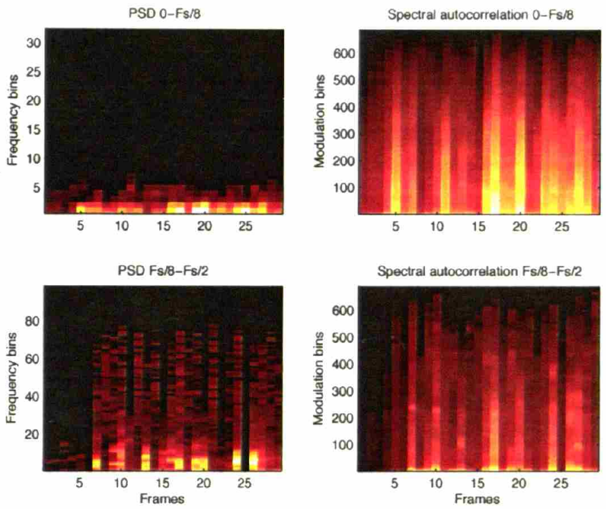


Figure 3-5: PSD and spectral autocorrelation frames at 0.5 Hz for $0 - \frac{F_s}{8}$ and $\frac{F_s}{8} - \frac{F_s}{2}$ for the first 60 seconds of "Shipbuilding."

behind this split was to model 'bass' type beat information (drums, bass guitar) vs. the rest of the spectrum. The song "Shipbuilding" begins with a piano vamp for a bar, followed by a slow drum pattern and bass. The low-end beatogram shows higher energies at the point where the drums stay stable, reflecting the modulations in the frequency domain.

3.2.3 Cepstral approaches

We now discuss two *cepstral* approaches, derived from speech research. Cepstral analyses are computationally cheap, well studied, and are a popular choice for music representations. [45]

Mel-frequency Cepstral Coefficients (MFCCs)

Mel-frequency cepstral coefficients (MFCCs) are defined as the mel-scaled cepstrum (the inverse fourier transform of the logarithm of the power spectrum on a mel scale axis) of the time-domain signal. They are widely used in speech recognizers and other speech systems as they are an efficiently computable way of reducing the dimensionality of spectra while performing a psychoacoustic scaling of frequency response.

To compute MFCCs from a (usually pre-emphasized) time domain signal x, the log power spectrum $P(x)$ is applied at some fixed windowing rate:

$$P_k = log||\mathcal{F}\{x\}|| \qquad (3.2)$$

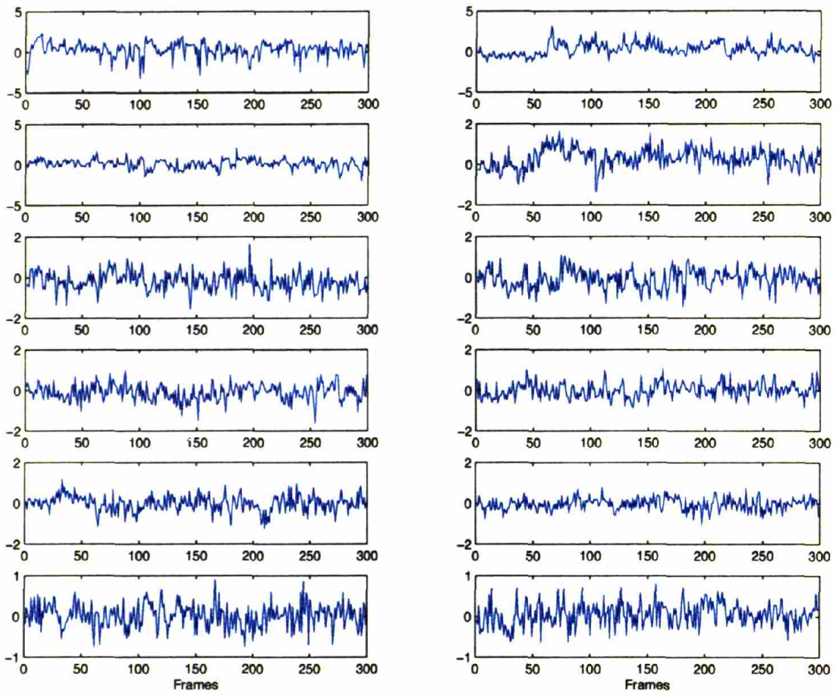


Figure 3-6: MFCC vectors 2-13 for the first 60 seconds of "Shipbuilding."

At this point the mel scale is applied, through integrating the frequency ranges of P. The mel scale was proposed in [76] from a listener study of pitch perception. A mel is a unit of pitch, and a mel of x is thought to appear twice as high as a mel of $\frac{x}{2}$. Starting with 1000 mel = 1000 Hz, listeners were asked to increase the frequency until they heard a tone twice the pitch of the original, and so on. The mel scale can be expressed as a weighting of frequency in Hz as:

$$mel(f) = 2595\, log_{10}(1 + \frac{f}{700}) \tag{3.3}$$

A plot of mel vs. frequency can be seen in Figure 3-7. After the power spectra is mel scaled (either through bin realignment or a filterbank) the cepstrum is computed either through an inverse fourier transform $\mathcal{F}^{-1}$ or the discrete cosine transform (DCT.) For computational efficiency, the DCT is often used since the input signal (the log mel scaled power spectra) is real and symmetric. The amount of mel filters in the frequency integration sets a maximum on the amount of mel cepstral coefficients that can be returned, but usually 13 are desired for most speech and music tasks.

Studies of MFCC for music analysis [45] [81] [10] have shown them to be useful for retrieval tasks. Specifically, in [45], a comparison between the de-correlating proper-

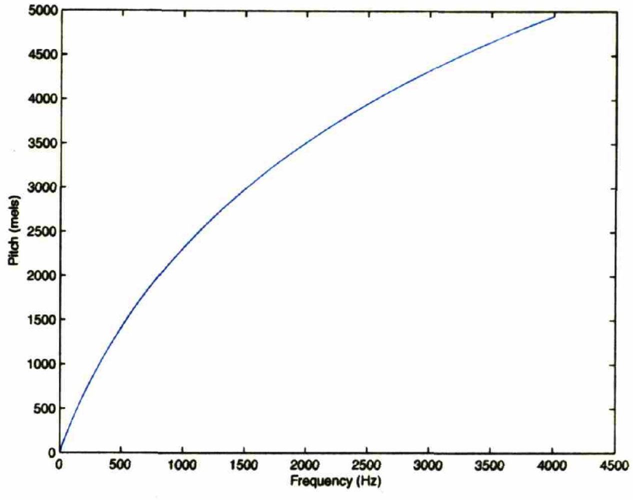


Figure 3-7: Mel scale: mels vs. frequency in Hz.

ties of principal components analysis (PCA) and the DCT step of the MFCC is made, showing them functionally equivalent for a speech / music discriminator. See Figure 3-6 for coefficients 2-13 of our test song.

Penny - Modulation Cepstra

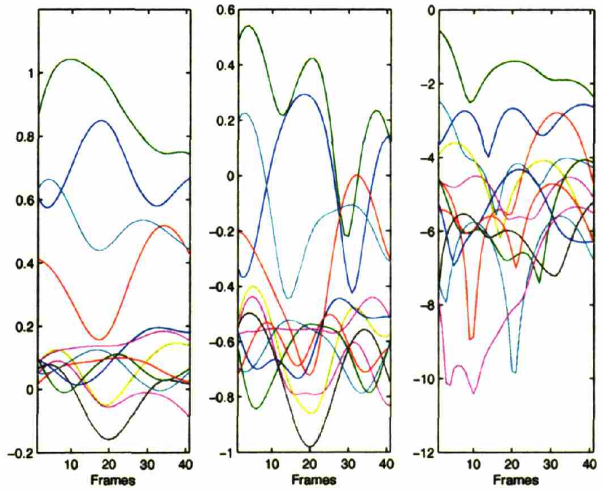


Figure 3-8: Penny V1, 2 and 3 for the first 60 seconds of "Shipbuilding."

Our feature space, nicknamed "Penny" is based on the MFCC, and could also be termed 'modulation cepstra:' Penny is the Fourier transform of the MFCC along time, with a weighting function to combine different cepstral coefficients. It is meant to capture cepstral rate changes along with the cepstra, thus including knowledge about time into the system. It is a cepstral analog to the 'beatogram' explained in Section 3.2.2.

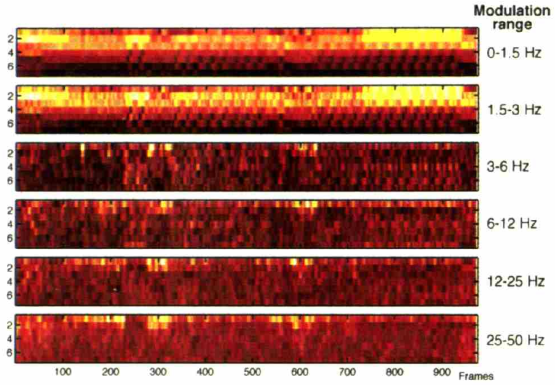


Figure 3-9: Six levels of structure are decoded for the song "A Journey to Reedham" by Squarepusher from the Penny feature.

To compute modulation cepstra we start with MFCCs at a cepstral frame rate (often between 5 Hz and 100 Hz), returning a vector of 13 bins per audio frame. We then stack successive time samples for each MFCC bin into 64 point vectors and take a second Fourier transform on these per-dimension temporal energy envelopes. We aggregate these results into 6 octave wide bins to create a modulation spectrum showing the dominant scales of energy variation for each cepstral component over a range of 1.5 Hz to 50 Hz (if the cepstral frame rate was 100 Hz.) The result is six matrices (one for each modulation spectrum octave) each containing 13 bins of cepstral information. The first matrix gives information about slow variations in the cepstral magnitudes, indicating things like song structure or large changes in the piece, and each subsequent matrix concentrates on higher frequencies of modulation for each cepstral coefficient. An example set of six matrices from the Penny analysis can be seen in Figure 3-9.

In practice, we found that using the first two matrices of Penny performed the best in music classification tasks. The high modulation data in the top four matrices lowered accuracy, perhaps at fault for representing inaudible or unimportant modulation response.

3.3 Evaluation

To get a feel for the different features' performance in a music retrieval task, we pitted Penny, MFCCs and PSD against each other in an artist identification task. (See Section

6.2.9 for more details.) We used an SVM (Section 5.2) with $C = 100$, 'auto-aux mode' on a gaussian kernel, (Section 5.3.2) and a maximum of 1,000 observations per class (chosen by pick-every-n) in a 1-in-20 artist ID problem. We used:

- Penny (first two matrices unrolled) at a 20 Hz cepstral frame rate
- MFCCs at a 20 Hz frame rate
- Penny (first two matrices unrolled) at a 5 Hz cepstral frame rate
- MFCCs at a 5 Hz frame rate
- MFCCs at a 5 Hz frame rate with delta embedding (computing the difference between adjacent frames and 'stacking' those results for a total of 26 dimensions per frame)
- PSD at a 5 Hz frame rate with $w = 256$.

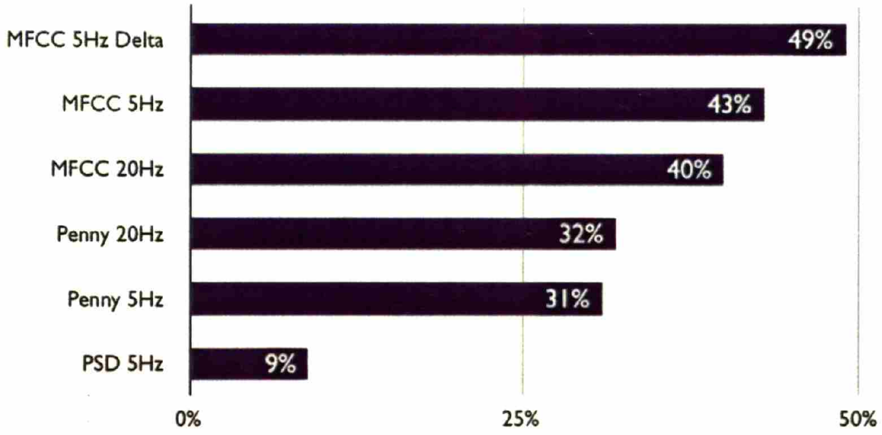


Figure 3-10: Evaluation of six features in a 1-in-20 artist ID task.

The results are shown in Figure 3-10: the percentage is based on how many songs were correctly classified (first choice) into the 20 artist bins on the test set. There were 200 songs total in the test set (10 song per artist), and 200 for training. We trained across albums if available. We see that the MFCC (best with delta embedding) outperforms Penny which vastly outperforms the PSD. Reading ahead to Chapter 6, it might be a surprise that we use the Penny feature in our meaning recognizers. There are two reasons that Penny is still a valuable feature for us: low data rate and time representation. Because of the overlap in the fourier analysis of the cepstral frames, the Penny data rate is a fraction of the cepstral rate. In usual implementation (Penny with a cepstral frame rate of 5 Hz, 300 MFCC frames per minute) we end up with 45 Penny frames per minute of audio. Even if MFCCs outperform at equal cepstral analysis rates, Penny needs far less actual data to make its classifications. This becomes more important in a meaning recognition context where each artist or song is represented by only 10 or 100 frames of information due to the amount of data. The time-awareness of Penny is also important to us as we wish to capture meaningful reactions such as 'fast' or 'driving,' which would not appear in non-periodic analyses (at low frequencies) such as the MFCC analysis.

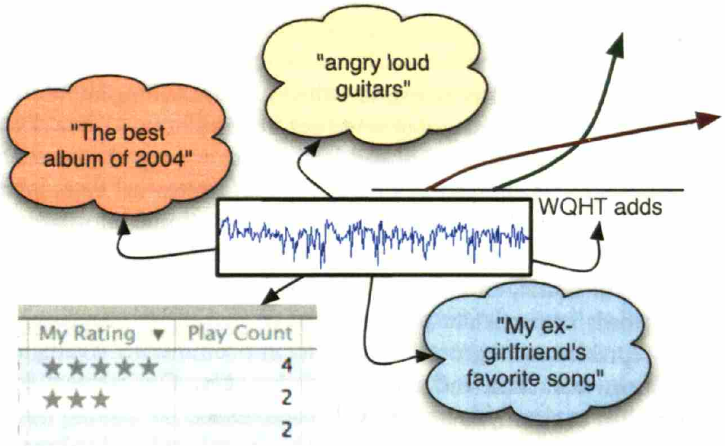
3.4 Conclusions

Throughout the meaning recognition component, we stick with Penny or modulation cepstra as it's an easily graspable concept with a simple implementation, high information content and musical applicability. We consider our design goals met: a musically informative feature that makes no assumptions and packs as much as it can into the dimensions.

CHAPTER FOUR
Contextual Analysis of Music

Just as the signal content of a piece of music needs to be packed in a form fit for later analysis, the extra-signal context associated with music needs to be identified and put into a robust representation. We call the contextual representation of music "community metadata," and it encompasses various types of data including free text, usage and correlation. In this chapter we formally define community metadata and describe ways of evaluating its utility.

4.1 Contextual Analysis for Music Understanding



My Rating ▼	Play Count
★★★★★	4
★★★	2
	2

Figure 4-1: Types of contextual information a signal can refer to: critical comment, community description, community usage, personal description, personal usage.

What should a contextual representation capture? Why is it worthwhile? Often, music is defined just as strongly by its signal as its relationship to other signals, which we can capture with similarity and co-occurence. Its effect, captured through usage and description, addresses meaningful reaction. And music's significance and correspondence can often be captured in critical description (record reviews, editorial information) or even its lyrics. Overall, a contextual representation should encompass everything about music that is missed in the signal. If we can't understand what a song is about and look it up on a metadata source such as All Music Guide [1] (AMG), it should be in a contextual representation. If an artist performs frequently in both a heavy metal and jazz group, that relation should be in a contextual representation. And a song played thirty times in the last week that hasn't been played before then should be noted in the contextual representation.

A larger question than 'why?' or 'what?' is *'how?'* There is a long history of getting computers to understand signals, and no clear analog to contextual information (which is a fuzzy concept to begin with.) Our main task is to first define what sort of data we *can* work with, and then figure out how to get it in a vector.

4.2 Community Metadata

We first defined community metadata in [83]. In our earlier work, we referred to community metadata as a 'cultural' representation, where 'cultural' was a way to refer to a community at large's opinion to a single point of reference (an artist.) Since then, we extended community metadata to various kinds of new sources, including usage data, playlists and similarity judgments. In this section we'll define community metadata as it relates to mined description, with additions in the form of new data sources.

4.2.1 Webtext

Our first goal is to model what people say when they are talking about music. To this end we try to capture a general gestalt reference from web mined data. Put simply, we crawl the web for music talk, parse it, and form it into a compact representation. This data mixes together and encapsulates many types of contextual data, including reaction, significance, editorial information, along with relational information. It should be noted that much like our Penny feature (Section 3.2.3) webtext is general, not specific to any type of music, and not with a predefined notion of structure. (We don't look for any words in particular or start from any set of web sources.) Our design goals were to represent some ground context M in a community k using unsupervised crawling, mining, extraction and clustering techniques. Our webtext representation can be applied just as cleanly to images, video and books.

Our input feature space for webtext community metadata comes from a a natural language feature extractor we developed for freeform web-extracted text. Our crawler takes as input a query term (usually an artist name, but also can work for album titles, etc.) which we augment with the search terms (domain dependent) such as 'music' and 'review.' For example, the 'review' search enhancement serves to limit the results to topical text about the artist (hopefully a review of an album, song, or concert.) Many results for the single-term only query 'Madonna,' for example, return splash pages or marketing concerns. The 'music' search enhancement similarly hopes to limit common-word artist names such as 'War' or 'Texas' to return only musically-related pages.

We send the query to a search engine[1] and then download a large amount of the top returned pages (this will be important later, we'll refer to the downloaded page count as p.) Each page in p is fed to a HTML parser that extracts the screen-viewable text. We then remove all extraneous whitespace and special characters and begin the process of feature extraction. We extract n-grams (sequences of ordered words having n words) for $n = 1$ (**n1** or unigrams) and $n = 2$ (**n2** or bigrams) from each page. We also feed the plain text input to a part-of-speech tagger (Brill's [13]), which fits each single word into a part of speech class (noun, verb, pronoun, adjective, etc.). Finally, we apply a noun phrase (NP) chunker (Penn's baseNP [58]), which selects terms to populate the **np** class.

Noun Phrases

Noun phrases	Not noun phrases
kittens	kittens went
angry guitars	guitars that become
the loud first gasp	loud first
not very funky music	funky

Table 4.1: An example of valid noun phrases and invalid noun phrases.

Noun phrases can be thought of as a noun extended with a maximal amount of descriptive text surrounding it. There is a defined grammar for noun phrase extraction, and once part-of-speech tagging has occurred, a simple rule-based NP chunker can operate on any amount of text. Noun phrases suggest more than a simple bi- or tri-gram since their content is limited to one idea. In the music domain, the sentence "Metallica employs screeching heavy metal guitars" leads to both 'metal guitars' and 'screeching heavy metal guitars' as noun phrases, but only the first is a possible bigram. Noun phrases can also serve as a simple noise reduction technique. A possible trigram from the above text could be 'employs screeching heavy,' which on its own does not provide much in the way of semantic description. But the NP extractor would retrieve the maximal NPs 'Metallica' and 'screeching heavy metal guitars,' as well as 'heavy metal' and 'heavy metal guitars.' The NP extractor would also return possibly musical problematic phrases like 'metal guitars' in a different context. Overall, the intuitive descriptive nature of noun phrases led us to believe that they should perform better than n-grams in the same retrieval or description task.

Adjective set

We also chose an adjectives-only subset **adj** of the **n1** class as a semantically descriptive feature set. The adjectives term set consists of every **n1** term tagged as an adjective by the part of speech tagger. The adjectives encapsulate a large amount of generalized de-

[1] We initially used Google^{TM}and later a larger variety of web service search engines

n2 Term	Score	np Term	Score	adj Term	Score
dancing queen	0.0707	dancing queen	0.0875	perky	0.8157
mamma mia	0.0622	mamma mia	0.0553	nonviolent	0.7178
disco era	0.0346	benny	0.0399	swedish	0.2991
winner takes	0.0307	chess	0.0390	international	0.2010
chance on	0.0297	its chorus	0.0389	inner	0.1776
swedish pop	0.0296	vous	0.0382	consistent	0.1508
my my	0.0290	the invitations	0.0377	bitter	0.0871
s enduring	0.0287	voulez	0.0377	classified	0.0735
and gimme	0.0280	something's	0.0374	junior	0.0664
enduring appeal	0.0280	priscilla	0.0369	produced	0.0616

Table 4.2: Top 10 terms of various types for ABBA. The score is TF-IDF for adj (adjective), and gaussian weighted TF-IDF for term types n2 (bigrams) and np (noun phrases.) Parsing artifacts are left alone.

scriptive content concerning the artists and are human-readable and understandable. For the entire list of unigrams, important descriptive terms tend to get lost among common words, technical terms, Internet-specific terms and typos. For applications such as query-by-description and description synthesis, the adjectives set is very useful. We also note that the adjective set is orders of magnitude smaller than the rest. The identified adjectives compose only about 1% of the unigrams found from our web crawls. An average adjective set for an artist is only 100 terms. The smaller number of terms helps speed learning and reduce complexity.

Artist terms

An important part of our feature space is the "artist term" set, **art**. We parse **n1** for terms that appear in the list of the top 6,000 artists found in our peer-to-peer crawling (see Section 4.2.2.) By doing this, we hope to be able to designate a section of our feature space to "similar artist" explanations. Many reviews of artists use other similar artists as touchstones to describe the music, and by creating a feature space that directly makes use of this, we may gain greater accuracy in our evaluation.

Although **n2** ends up performing best (alone) in a similarity evaluation, we tend to use in practice either **adj** or **np** for specific meaning recognition tasks. **adj** gives us 'filters,' terms that describe effects of music such as 'loud' or 'romantic' while **np** gives us 'events:' terms that denote something happening in the music ('heavy metal guitars) or specific information about the music (artist names, song titles.)

Scoring

After extracting the terms from our webtext mining, we now look to representing the salience of each term. Salience is the importance of a term t given the ground context M. It is explicitly ***not*** the probability of a term given a context: $P(t|M)$, as this would only lead to optimizing specific terms appearing in the music context (band members' names, song titles) Rather, the salience should reflect value in a summarized version of M (i.e. a test should indicate that the topmost salient terms for M would adequately describe M,) and to achieve this we need to slightly modify the usual information retrieval task.

After extracting the features, we can compute term frequency and document frequency for each term type in each context set. Term frequency (f_t) was defined as the percentage of retrieved pages that contained the given term (treating each retrieved page in p separately). Document frequency (f_d) was computed across the entire retrieved set, treating each entire context as a document. We treat both f_t and f_d as a normalized probability between 0 and 1 (where f_t is $P(t|M)$ and f_d is $P(t|M^{\infty})$), and then compute the TF-IDF (f_t/f_d) [66] value of each term, which we also normalize between the local minimum and maximum values for each artist.

If two contexts share a term in their feature space, we say that those terms overlap with an associated salience. The scores for overlap are accumulated to create a numerical similarity metric between two contexts. We compute overlap for all term types that we have extracted. To compute the score of two terms having overlap, we experimented with various thresholding and smoothing metrics. The score of an overlap could simply be 1 (a match of a term on two contexts) or it could a function of the term and/or document frequency. In the former case, common words such as 'music' or 'album' get very high overlap among all contexts, and typically do not retrieve musically intelligent terms. Considering this, we use a metric that is based on the TF-IDF value of the term in question. The nature and size of the n1, n2 and np sets (in the tens of thousands for each context) led us to believe that we needed a way to emphasize the terms found in the middle of the span of IDF values. The intuition is that very rare words, such as typos and off-topic words rarely used on music pages, should be down-weighted in addition to very common words such as 'the'. To achieve this, we used a gaussian smoothing function that, when given appropriate μ and σ (mean and standard deviation) values, can down-weight both very common and very rare terms to create a salience function $s(t, M)$:

$$s(t, M) = \frac{P(t|M)e^{-(log(P(t|M^{\infty}))-\mu)^2}}{2\sigma^2} \tag{4.1}$$

where $P(t|M^{\infty})$ is renormalized such that the maximum is the total document count. Through an empirical evaluation on an artist similarity task (shown below) we ended up choosing 0.9 for σ and 6 for μ. To compute an overlap score, we simply add the gaussian-weighted result for each term found in both the comparison and the base artists' sets.

Evaluating Webtext

To see which forms of webtext were more apt at understanding the aboutness of their context, we performed an artist similarity evaluation. In these experiments, we chose artist names (414 artists) for our M. Our experiments concentrate on evaluating the fitness of our representation by comparing the performance in computing artist similarity with an edited collection. We note that our representation is suitable for many tasks; but artist similarity is well-posed and we can perform formal evaluation with "ground truth" data.

During each of the following experiments, we ran a system that computes overlap of terms. Our grounding assumption is that similar artists share features in our space, and that our representation allows for enough generality to classify artists into similar clusters. To evaluate, we compare the performance of our varying feature types in the task of predicting the All Music Guide's [1] similarity lists (for each of our 414 artists, AMG on average lists 5 other artists also in our set that are known similar).

For each artist in our set, we take the top n terms from their feature space. n is defined as a rough minimum for the size of the feature space; we want each artist to have the same amount of terms for comparison purposes. For the **n1** term type, for example, n is 1000 (**n2**: n=5000, **np**: n=5000, **adj**: n=50, **art**: n=500). The top n terms are sorted by the overlap scoring metric, either using the gaussian-weighted TF-IDF (Equation 4.1) with $\mu = 6$ and $\sigma = 0.9$ or the 'flat' TF-IDF alone. We then compare this feature space against every artist in the current artists' edited similarity list. The overlap scoring metric is averaged for each similar artist. We then do the same for a randomly chosen set of artists. If the overlap score is higher for the similar artist set, we consider that our feature space correctly identified similar artists. The percentages shown below indicate the percentage of artists whose similar cluster was predicted. We expect this task to be relatively easy, i.e., we expect percentages $\gg$ 50%. Note although that the entire set of artists (which correlates with the interests of OpenNap users from our peer-to-peer crawling in Section 4.2.2) is predominately rock and pop with few artists from other styles of music.

We also compute a more powerful metric which we call *overlap improvement*, which is the ratio between overlap scores for similar artists compared to randomly chosen artists. A higher overlap improvement indicates a stronger confidence of the feature space for this task.

	n1	n2	np	adj	art
Accuracy	78%	80%	82%	69%	79%
Improvement	7.0×	7.7×	5.2×	6.8×	6.9×

Table 4.3: Results for the flat TF-IDF scoring metric for artist similarity.

	n1	n2	np	adj	art
Accuracy	83%	88%	85%	63%	79%
Improvement	3.4×	2.7×	3.0×	4.8×	8.2×

Table 4.4: Results for the gaussian-weighted TF-IDF scoring metric for artist similarity.

We see in the results of Table 4.3 and Table 4.4 (and as a chart in Figure 4-2) that the gaussian weighted TF-IDF outperforms the unweighted TF-IDF for a few key term classes in accuracy. The improvement metric is less obvious, however, but explained by the lower scale of the gaussian weighted TF-IDF, and the poor normalization of the flat TF-IDF. Overall, however, the results are promising: using contextual features alone, we can predict with high accuracy similar artist clusters.

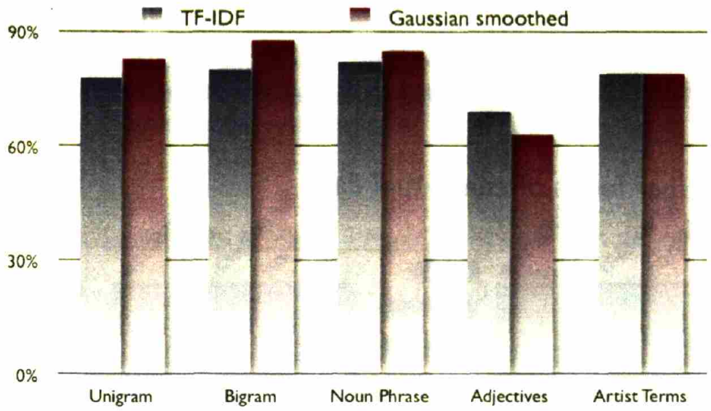


Figure 4-2: Results for both gaussian-weighted and flat TF-IDF accuracy in an artist similarity task.

4.2.2 Peer to Peer Usage

We also created a similarity measure of artists based completely on user collections in peer to peer networks. We defined a collection as the set of artists a user had songs by on their shared folder during a crawl of the OpenNap peer-to-peer music sharing network. If two artists frequently occur together in user collections, we consider them similar via this measure of community metadata. We also define a collection count $\mathcal{C}(artist)$ which equals the number of users that have *artist* in their set. $\mathcal{C}(a, b)$, likewise, is the number of users that have both artists a and b in their set.

However, one particular problem of this method is that extremely popular artists (such as Madonna) occur in a large percentage of users' collections, which down-weights similarity between lesser-known artists. We developed a scoring metric that attempts to alleviate this problem. Given two artists a and b, where a is more popular than b (i.e., $\mathcal{C}(a) \geq \mathcal{C}(b)$), and a third artist c which is the most popular artist in the set; they are considered similar with normalized weight:

$$\mathcal{S}(a, b) = \frac{\mathcal{C}(a, b)}{\mathcal{C}(b)}(1 - \frac{|\mathcal{C}(a) - \mathcal{C}(b)|}{\mathcal{C}(c)}) \qquad (4.2)$$

The second term is a popularity cost which down-weights relationships of artists in which one is very popular and the other is very rare. We evaluated this similarity metric by replacing the All Music Guide ground truth with this data on the same experiments as in Section 4.2.1. The results are in Table 4.5.

	n1	n2	np	adj	art
Accuracy	80%	82%	84%	68%	72%
Improvement	2.6×	2.1×	2.4×	7.1×	4.5×

Table 4.5: Similarity accuracy using OpenNap community data as ground truth.

4.2.3 Other forms of contextual data

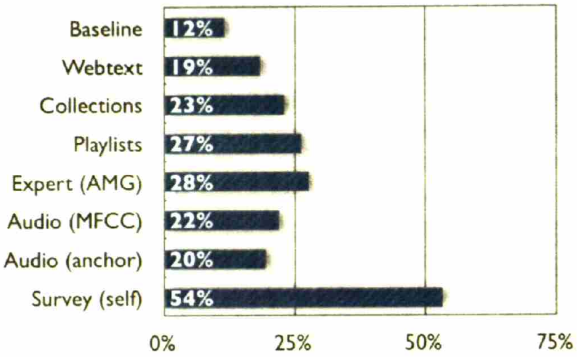


Figure 4-3: Top-rank similarity agreement against the Musicseer [23] similarity survey among different types of signal data and community metadata. [10]

We investigated other forms of contextual data for the meaning recognition, which we will briefly mention here. In [23] we created the 'Musicseer' game, which asked users to rate artist similarity directly or through an 'erdos distance' metric. Using that data as ground truth, we evaluated self-similarity (using the user data as extracted community metadata to test against itself), edited artist similarity metadata (from AMG), collection data as extracted from the OpenNap analysis (Section 4.2.2) and user submitted playlist data from the popular web site Art of the Mix [2]. A large scale evaluation of these contextual sources (along with two audio-derived features) was performed in [10], using top rank agreement to the Musicseer rankings as a fitness metric.

The results from this analysis is in Figure 4-3. The baseline results were from random similarity, the 'self' results indicate inter-rater reliability between all survey respondents (i.e. the respondents agreed with themselves 54% of the time.) The picture as painted should be clear: there is no single type of contextual or signal derived data that solves the 'meaning problem.' By considering all possible forms together we hope to capture as much contextual data as we can, and let the recognition system sort out what is important.

4.3 Community Modeling

An immediate problem with extracting contextual data from the internet as in our webtext analysis is that there is no notion of intra-domain meaning specificity. That is, we assume that the salience of 'romantic' to a context like "Barry White" is agreed upon as the mean of the entire world's salience. In reality, there are many meanings of romantic and in particular many views on how romantic Barry White is. We want to create multiple sub-contexts within each ground context M, and we need to do this without any explicit survey or heuristic measure.

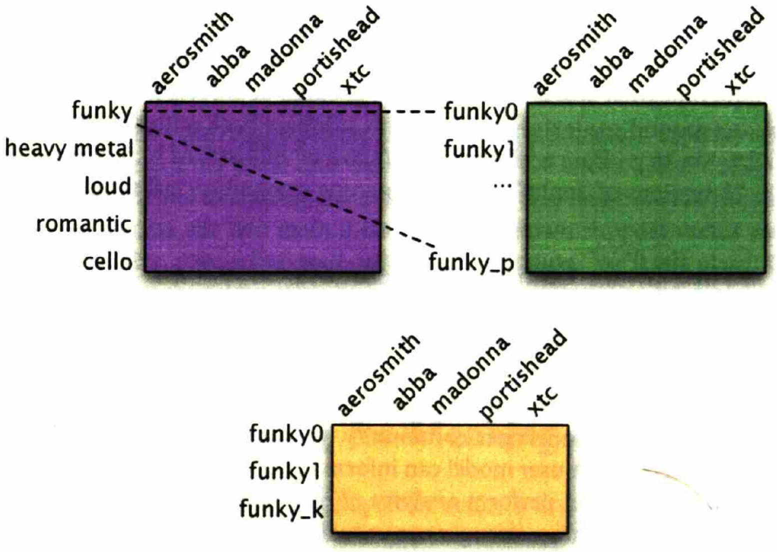


Figure 4-4: Community modeling process: expanding a ground context vs. term matrix to the sub-contexts, and then reducing the p sub-contexts into k community views on a single term.

The problem becomes: given a ground context M (an artist name, album title, song) and a set of p returned pages each with a group of terms t each with associated salience $s(t, M)$, how can we compute a model of salience given a sub-context: $s(t, M)_{1...k}$? Thanks to the large amount of p stored for each M (usually between 50 and 200) and the variety of internet sites with their own communities and language patterns, we can cluster the ground context into k communities without any domain knowledge or demographics. We rely on the assumption that there are extractable differences in description among those p pages, and over a large enough set, correspondences among p will form communities.

The actual implementation relies on singular value decomposition (SVD) (see Section 6.2.4 for more details.) When SVD is applied to text stores such as ours, it is often called Latent Semantic Analysis or Latent Semantic Indexing. However, our aim is not to reduce the dimensionality of the term space, as LSI methods do, but rather to reduce the dimensionality of the sub-contextual space, the pages p. We assume that at the outset each of these extracted p per ground context M represents their own community. This is overall not true, so we rely on the SVD to reduce the rank of this space from p to k.

Implementationally, we collect the webtext data as usual but do not bag all the returned pages p into the same $s(t, M)$ equation as usual, but rather create a $s(t, M, p)$ function from each p. We then arrange this new data as t matrices of size $p \times M$, one matrix for each term. Each matrix represents view of page p for ground context M of its term t. We'll call this matrix $\mathbf{R}$, or the sub-context vs. context space.

We extract the k top principal components using PCA (Section 6.2.4) of $\mathbf{R}$, which results in a $k \times M$ matrix for each term in t. The PCA step reduced the dimensionality of the sub-contextual space from p to k by looking at correspondences of the target label across ground contexts. Projecting this subspace back onto the entire contextual space allows us to go from a 'global' $s(t, M)$ to a set of k $s(t, M)_{1...k}$. We can treat this space as "k versions of 'funky'" for later meaning recognition tasks if needed.

4.3.1 Time-aware Community Metadata

We note that the contextual data extracted in community metadata has an important benefit in that it can be crawled or extracted at regular intervals, this 'time-aware' quality of the representation is a clear advantage over signal-based analyses. Just as we store sets of k $s(t, M)_{1...k}$ models per community, we can also store a memory of recent contextual perception. A user model can inform which point in time the listener wants to relate to. We can also perform analyses of community perception over time, and integrate those results into a stronger contextual representation.

4.4 Conclusions

By looking at web-mined description, peer to peer usage patterns and other globally-available data, we can reliably show that we are extracting necessary contextual information about music. Our later experiment make heavy use of linking the contextual description to the signal, but the types of contextual community metadata described here can be used for many tasks alone, without any perception involved.

We note our largest problem is one of contextual scale: through the process of our data collection, most of our context is at the artist level. This becomes problematic when linking song or sub-song level acquired description ("This break is funky," or "This song is sad, but the band is usually happy") to audio signals. Much of the contextual scale issues are considered part of future work: our goal here is to generate a generalized representation of aboutness given a large set of data.

CHAPTER FIVE
Learning the Meaning

Now that we know how we're going to represent music (modulation cepstra) and how we're going to represent the contextual extra-signal part of music (community metadata) we now need a way to actually learn the meaning (the link between the two.) In this chapter we discuss our two main tools, the well-known support vector machine (SVM) and its relative, regularized least-squares classification. We'll talk about the design goals for learning the relationship between high dimensional audio data and up to 200,000 output classes and how to implement this problem at a large scale.

5.1 Machine Learning for Meaning Recognition

The machine learning algorithm will hopefully create a link between the perception and the interpretation. It should be able to create 'machines' that can classify new perception offline (after training is completed.) We would like to train a single machine for each particular type of interpretation. This includes descriptors ("loud," "romantic," "funky"), events ("beating drums," "first loud burst of noise,") and anything else we can think of. However, the problem has three important caveats that separate it from most classification tasks:

- **Too many output classes**: Each audio frame can be related to up to 200,000 terms (in the unconstrained case.) Most contexts have community metadata vectors of 10,000 terms at one time. For a standard machine learning technique, this would involve costly multi-class learning.
- **Classes can be incorrect or unimportant**: Due to the unsupervised and automatic nature of the contextual feature extraction, many are incorrect (such as when something is wrongly described) or even unimportant (as in the case of terms such as 'talented' or 'cool' – meaningless to the audio domain.) We would need a system that could quickly fetter out such errant classes.
- **Outputs are mostly negative**: Because the decision space over the entire artist space is so large, most class outputs are negative. In an example 51 artist set, only two are described as 'cynical' while 49 are not. This creates a bias problem for most machine learning algorithms and also causes trouble in evaluation.

One possible way to learn this relation is to train a binary classifier on each possible interpretation, given the audio frames as input examples. However such training has a large startup time for each new class. We show below a novel algorithm that eliminates this startup time and allows for multiple classes to be tested easily.

5.2 Support Vector Machines

Support vector machines (SVM) [80] [14] are a classification technique based on structural risk minimization (SRM), where the lowest probability of error given a hypothesis is found. The output of the SVM is a set of support vectors **w** and a classification function given data in **x** and a bias term b:

$$f(x) = \text{sgn}\left(\langle \mathbf{w}, \mathbf{x}\rangle + b\right) \tag{5.1}$$

The problem can be viewed as form of Tikhonov regularization [26], finding the best function f in the hypothesis space $\mathcal{H}$

$$\min_{f\in\mathcal{H}} \frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i, f(\mathbf{x}_i)) + \lambda\|f\|_K^2 \tag{5.2}$$

where $V(f(x), y)$ is a *loss function* that shows the loss (or 'price') when, given x, we say $f(x)$ and the ground truth is actually y. **x** contains the observation space of ℓ frames or observations. If the chosen loss function is a *hinge loss*,

$$V(f(\mathbf{x}), y) = \max(1 - yf(\mathbf{x}), 0) \tag{5.3}$$

the regularization problem now is

$$\min_{f\in\mathcal{H}} \frac{1}{\ell}\sum_{i=1}^{\ell}(1 - y_i f(\mathbf{x}_i)) + \lambda\|f\|_K^2 \tag{5.4}$$

Because the hinge loss function $V(f(x), y)$ is not differentiable adding in a necessary slack variable ξ makes the new regularization problem

$$\min_{f\in\mathcal{H}} \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i + \lambda\|f\|_K^2 \text{ where} \tag{5.5}$$

$$y_i f(\mathbf{x}_i) \geq 1 - \xi_i, \text{ and } \xi_i \geq 0 \tag{5.6}$$

5.2.1 Representer Theorem

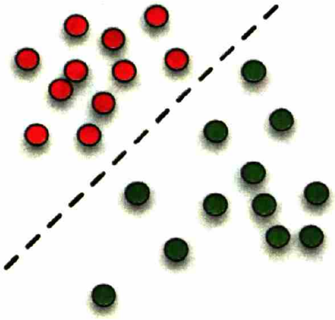


Figure 5-1: The optimally separating hyperplane of two classes, where the data is linearly separable.

The usefulness of the SVM lies in the *representer theorem*, where a high dimensional feature space $\mathbf{x}$ can be represented fully by a generalized dot product (in a Reproducing Kernel Hilbert Space [7]) between $\mathbf{x_i}$ and $\mathbf{x_j}$ using a *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j)$. For example, the binary classification problem shown in figure 5-1 could be classified with a single linear hyperplane learned by an SVM. However, non-linearly separable data as in Figure 5-2 need to consider a new topology, and we can substitute in a gaussian kernel function that represents data as

$$K_f(x_1, x_2) = e^{-\frac{(|x_1 - x_2|)^2}{\sigma^2}} \tag{5.7}$$

where σ is a tunable parameter. Kernel functions can be viewed as a 'distance function' that compares all the high-dimensionality points in your input feature space and represents all your data as some distance between points. There is a bit of engineering in choosing the best kernel function, as the function should reflect structure in your data, and later we discuss an alternate kernel for music analysis. For now it should be noted that kernel matrices (the $\ell \times \ell$ matrix $\mathbf{K}$ that contains all the kernel evaluations) should be symmetric positive semi-definite; that is, all the eigenvalues of $\mathbf{K}$ are non-negative.

If we substitute $f(x)$ with the new $f^*(x)$

$$f^*(\mathbf{x}) = \sum_{i=1}^{\ell} c_i K(\mathbf{x}, \mathbf{x}_i) \tag{5.8}$$

we end up with the 'primal' SVM problem, a constrained quadratic programming problem:

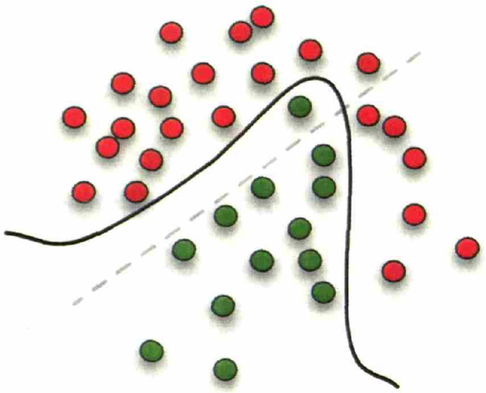


Figure 5-2: The generalized optimally separating linear hyperplane of two classes (dashed line) along with the optimally separating gaussian radial basis hyperplane.

$$\min_{\mathbf{c}\in\mathbb{R}^\ell,\xi\in\mathbb{R}^\ell} \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i + \lambda\mathbf{c}^T K\mathbf{c} \text{ where} \tag{5.9}$$

$$y_i(\sum_{j=1}^{\ell} c_j K(x_i,x_j)+b) \geq 1-\xi_i \text{ and } \xi_i \geq 0 \tag{5.10}$$

(Here K is the filled-in kernel matrix, not the kernel function.) It should be noted that through Lagrangian analysis the primal is presented as a dual program that is easier to solve. In practice, refer to [14] for more information.

λ is a 'regularization parameter' often in practice re-appropriated as C as a tunable training parameter:

$$C = \frac{1}{2\lambda\ell} \tag{5.11}$$

C is colloquially the 'generalization knob' as it affects the tradeoff between accuracy in classification given the observations in $\mathbf{x}$ and generalization to new data. Substituting in the new definition of C and relating the kernel vectors to the support vectors leaves us with the new primal of

$$\min_{\mathbf{w}\in\mathbb{R}^n,b\in\mathbb{R}} C\sum_{i=1}^{\ell}\xi_i + \frac{1}{2}\|\mathbf{w}\|^2 \text{ where} \tag{5.12}$$

$$y_i(\langle\mathbf{w},\mathbf{x}\rangle + b) \geq 1-\xi_i \text{ and } \xi_i \geq 0 \tag{5.13}$$

Where the $f(x)$ of new data can be predicted once $\mathbf{w}$ is found using Equation 5.1. Note that the generalized dot product used in the primal program must also be used to classify new data.

5.3 Regularized Least-Squares Classification

Regularized Least-Squares Classification (RLSC) is a powerful approach to solving machine learning problems [62]. It is related to the Support Vector Machine in that they are both instances of Tikhonov regularization, but whereas training a Support Vector Machine requires the solution of a constrained quadratic programming problem, training RLSC only requires solving a single system of linear equations. Recent work [30], [62] has shown that the accuracy of RLSC is essentially identical to that of SVMs.

Starting with the regularization problem in Equation 5.2, we substitute the square loss for $V(f(\mathbf{x}, y))$:

$$V(f(\mathbf{x}, y_i) = (f(\mathbf{x}) - y)^2 \tag{5.14}$$

which makes the problem now

$$f = \arg\min_{f \in \mathcal{H}} \frac{1}{\ell} \sum_{i=1}^{\ell} (f(x_i) - y_i)^2 + \lambda \|f\|_K^2 \tag{5.15}$$

Using the representer theorem, we can again repurpose the regularization problem using Equation 5.8. However, if we keep the entire kernel matrix K, we can classify data with

$$f(x_i) = (K\mathbf{c})_i \tag{5.16}$$

It would be helpful at this stage to refer to K as your observations (run through the kernel space) and $\mathbf{c}$ as your 'machine' that transforms a point of data into the output of the function. Using the representer theorem, our regularization problem is now

$$f = \min_{f \in \mathcal{H}} \frac{1}{\ell} (K\mathbf{c} - y)^2 + \lambda \|f\|_K^2 \tag{5.17}$$

and knowing that our function $f(x)$ is in the form of Equation 5.8, we now need to minimize a function $g(\mathbf{c})$ to find $\mathbf{c}$:

$$g(\mathbf{c}) = \frac{1}{\ell} (K\mathbf{c} - y)^2 + \lambda \mathbf{c}^T K\mathbf{c} \tag{5.18}$$

which, after taking the derivative with respect to c and setting the equation 0, we arrive at

$$\mathbf{c} = (K + \lambda \ell I)^{-1} y \tag{5.19}$$

where I is the identity matrix. Remembering the definition of λ from Equation 5.11, we are left with

$$(K + \frac{I}{2C})\mathbf{c} = \mathbf{y}, \tag{5.20}$$

Since C is a constant that in practice remains the same with different problems, we usually ignore the 2 in the denominator.

5.3.1 Multiclass RLSC

A key property of this approach is that in Equation 5.20 the solution c is *linear* in the right-hand side y. We compute and store the inverse matrix $(K + \frac{I}{C})^{-1}$ (this is numerically stable because of the addition of the regularization term $\frac{I}{C}$), then for a new right-hand side y, we can compute the new c via a simple matrix multiplication.

For example, given a fixed set of training observations, we can create the kernel matrix **K**, add the regularization term, and invert. (In practice we use iterative optimization techniques.) To create machines for each possible output class, simply multiply the inverted matrix by a truth y vector (where y is $-1 \ldots 1$ for each observation in $1 \ldots \ell$. The resultant c will be able to classify new data by projecting a test point $\mathbf{x}_{\mathbf{test}}$ through the kernel function K and then c:

$$f(\mathbf{x}_{\text{test}}) = \mathbf{c}K(\mathbf{x}_{\text{test}}, \mathbf{x}) \tag{5.21}$$

in implementation, this can be interpreted as your classifier multiplied by every point of training data evaluated through the kernel against your new test point. In effect, RLSC can be seen as equivalent to an SVM in which every training observation becomes a support vector. The vector c for each class weights each observation's importance to the resultant classifying function.

In most multi-class problems, training time is linear in the amount of classes n. Training an SVM to discriminate amongst n classes either requires n SVMs in a one-vs-all scheme (in which each SVM is asked to discriminate between membership in their class or no membership,) or up to $(n * (n - 1))$ SVMs in a one-vs-one scheme (1 vs. 2, 2 vs. 3, 3 vs. 4, 4 vs. 5, 1 vs. 3, etc.) [55] For RLSC, since adding new classes is implemented with a simple matrix multiplication, otherwise intractable problems are now possible.

5.3.2 RLSC Optimizations

RLSC does have the downside of being memory-hungry: storing the entire **K** in memory or disk becomes unwieldy soon. 10,000 observations stored in a full matrix using double-precision floating point numbers requires 800 megabytes of storage. To implement RLSC we investigated two short-circuit paths for a large-scale implementation, as well as a method of estimating the kernel parameter σ. In our case we fully solve the system of equations, we should note that there exist ways of iteratively optimizing the RLSC inversion using conjugate gradient methods [32].

Cholesky Decomposition

The Cholesky decomposition [32] is a well known method of matrix factorization meant as a front end to full inversion. It allows inversions of positive definite matrices to be computed in half the operations over Gaussian elimination, and in implementation only requires the lower triangle of the matrix to be stored. The inverse of the kernel matrix $\mathbf{K}$ (which by definition is symmetric positive semidefinite, the $\frac{I}{C}$ term makes it fully definite) is

$$\mathbf{K}^{-1} = (\mathbf{L}\mathbf{L}^{\mathbf{T}})^{-1} \tag{5.22}$$

where $\mathbf{L}$ was derived from the Cholesky decomposition. There are highly tuned algorithms for both computing the Cholesky decomposition in place on a lower-triangular matrix and also the inverse of the Cholesky factorization available in LAPACK [6].

In our implementations, we use the single precision LAPACK Cholesky (SPPTRF) and inverse (SPPTRI) on a packed lower triangular matrix. This halves the memory constraints (only $\frac{\ell^2}{2}$ 4-byte floats are stored and is roughly twice as fast as a full inversion.

Parallel RLSC

Our work is computed on a cluster of machines, each with two processors. RLSC solving is both memory and processor bound, so the collection of multiple machines, each with their own memory subsystems, is a boon to solving very large RLSC problems. To solve a 'parallel RLSC' problem we first split our observation space in ℓ into t slices, in which the kernel evaluations are all only among their own slice. For example, instead of having an (currently infeasible) 50,000 × 50,000 matrix (5 gigabytes) we partition the data among 10 processors randomly, and each node in t receives an observation space of $\ell = 5000$. This randomized subsampling is a 'curse of dimensionality' fix for the observation space and increases accuracy over the small single node results in an artist identification task.

Auto-aux Mode

The variance parameter σ in the gaussian kernel (Equation 5.7) is often an important optimization in the learning process. σ needs to reflect the scale of the data, and is normally set to $\sigma = 0.5$ if the data has been pre-normalized. But in our task we combine features from different sources and we've found that normalization has a harmful effect on testing accuracy. To that end, we created a quick heuristic estimation of σ, where $\mathbf{A}$ contains your observation vectors:

$$\sigma \approx \sqrt{\max_{i\in d, j\in \ell}(\mathbf{A}_{ij})} \tag{5.23}$$

We find with this simple approximation that training on non-normalized data with a gaussian kernel approaches the performance of a hand-tuned (or derived through cross-validation) σ without the computation cost. For multi-class tasks, σ can be estimated per class instead of over the whole dataset, which has a net positive effect on multi-class scaling.

5.4 RLSC vs. SVM

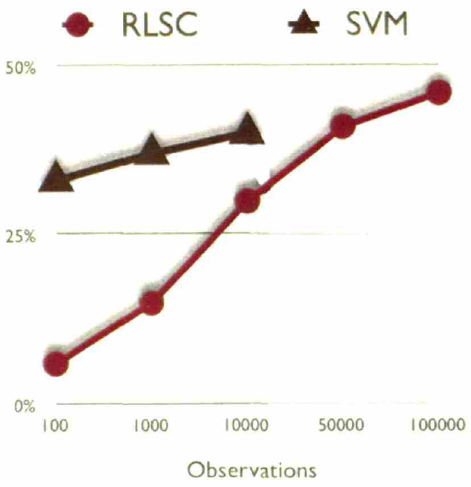


Figure 5-3: RLSC vs. SVM for accuracy in a 1-in-20 prediction task. Last two SVM datapoints not computed.

The square loss function of RLSC makes it a much simpler process that handles multiclass learning more cleanly, but we are concerned if it performs adequately against the SVM. To evaluate this performance, we pitted RLSC against SVM in an artist identification task (covered in more detail in Section 6.2.9.) Artist ID is a hard music-IR problem with solid ground truth and a strong reliance on features and learning algorithm. We chose a baseline feature space (MFCCs) and ran both on the same 1-in-20 artist ID problem. We varied the amount of observations (ℓ), kept C at 100, used a gaussian kernel (Equation 5.7) with 'auto-aux' mode (described in Section 5.3.2) for the σ kernel parameter. Our SVM solver was Nodelib [27] and we used our own RLSC solver. We should note that the observation count for the SVM is *per-class*, that is for each 1 vs. all classifier it loads ℓ in to learn the model. For RLSC, since it is learning all classes at once, the ℓ is over all the classes.

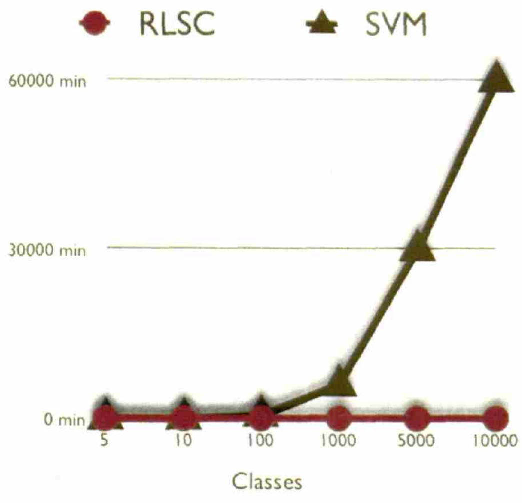


Figure 5-4: RLSC vs. SVM for compute time, number of classes variable. Last two SVM datapoints simulated.

The accuracy results are in Figure 5-3. With small observation counts RLSC does not fare as well as the SVM mostly due to the low observation count per class. But at significant counts (10,000) the RLSC approaches the SVM and continues to climb with more observations. We did not stick around for the SVM to finish computing 50,000 observations per class or above. We note the simulated memory (Figure 5-5) and processor time (Figure 5-4) comparisons for the two algorithms: RLSC is memory dependent on observation count, where SVMs normally look at small pieces of the data at a time in a cache, so the memory use stays static. For computation time the SVM is linearly dependent on class count, while RLSC stays static independent of class count.

5.5 Conclusions

Our take-home message is to show that RLSC is not fit for binary or small multi-class problems but for larger multi-class problems the accuracy matches and can even eclipse that of the SVM, and can be easily and efficiently implemented. The SVM is simply not applicable to our large in-use test case of up to 10,000 output classes operating on over 200,000 ℓ.

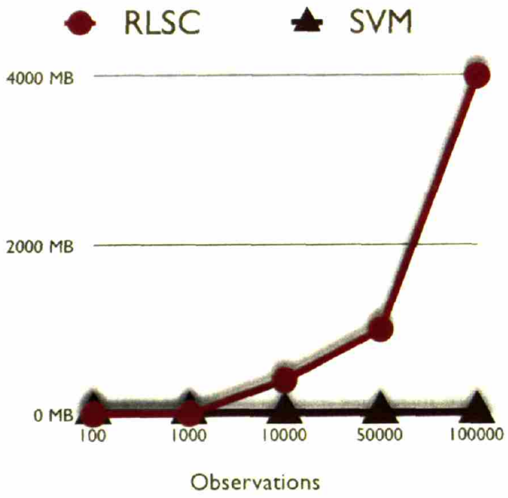


Figure 5-5: RLSC vs. SVM for memory / disk allocation, number of observations variable.

CHAPTER SIX
Semantic Basis Functions

In this chapter we'll evaluate our meaning recognition system in two separate forms: first through the lens of "query-by-description," in which a predictive model of community supplied reaction applies to new 'unheard' data. We'll discuss the reasoning behind this evaluation along with two applications of this type of evaluation: parameter learning and textual analysis in the form of a bi-corpus of record reviews. We then discuss our more robust evaluation in the concept of 'semantic basis functions–' meaning classifiers which predict types of meaning (mostly relational and reaction) from perception and represent any new perception by a linear combination of this reaction– and how meaning can directly influence music and other multimedia understanding tasks for better accuracy in classification.

6.1 Query By Description

We test the strength of our perception to interpretation models by an on-line 'query-by-description' (QBD) task, in which the system is asked to label as-yet-unheard music, and we use prediction accuracy to evaluate the models. The work in query-by-description has two direct uses: first as an interface ("play me something romantic") and second as an analysis method to detect musicality of terms. Our evaluation framework allows us to detect which terms have naturally higher correlations to audio than others. Words such as 'funky' have high predictive accuracy, while words like 'bad' or 'sexy' do not. This analysis provides an insight into the meaning of music description and can be used for numerous text and audio understanding tasks.

6.1.1 Term prediction

To compute a predictive model $\mathbf{c}_t$ for each term t, we use RLSC to learn the relationship between the audio features and each term in the community metadata vector. For this type of experiment, we use artists as our ground context: each term will be related at the artist level, and the audio data will be culled randomly from any of that artists' music. This represents an enormous chasm of scale: terms in free description can relate to all kinds of structure, from the artist and its community to the notes and samples in their recordings. Our intuition is that for this type of experiment, faced with thousands

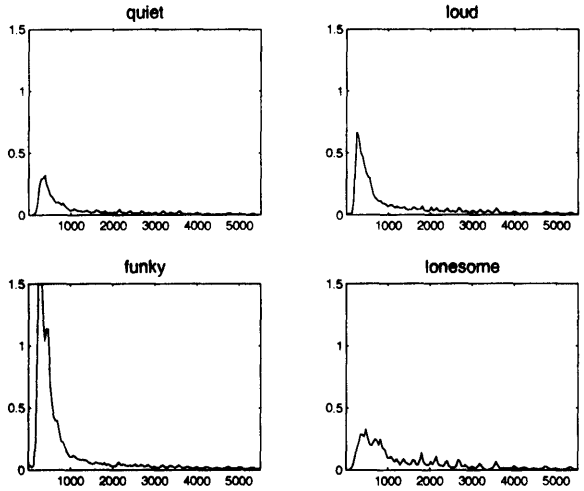


Figure 6-1: Mean spectral characteristics of four different terms evaluated from a query by description task. Magnitude of frequency on the y-axis, frequency in Hz on the x-axis.

of artists and tens of thousands of target classifiers, we are up against more than issues of scale in semantic attachment. Our hope is that the evaluation component of the learning will lead us to tune our models and be able to tell which types of classifiers are in fact learning a strong connection between the perception and interpretation.

After choosing a set of a artists, we choose $\frac{\ell}{a}$ random frames of audio from each, where ℓ is our total amount of frames to consider over the entire term prediction set. Using the gaussian kernel process described in Chapter 5, we create the full kernel matrix $\mathbf{K}$, stored either in full or distributed across processors, and add the $\frac{\mathbf{I}}{C}$ term, usually choosing a C of 100 and the 'auto-aux' (Section 5.3.2) mode for σ. We then invert the matrix using the Cholesky decomposition and store the result. This becomes our support vector matrix $\mathbf{S}$.

For each term in t, to create the $\mathbf{c}_t$ we simply multiply a ground truth vector y_t by $\mathbf{S}$. y_t is an ℓ-long vector, normally containing the $s(t, M)$ salience metric outlined in Chapter 4: the $s(t, M)$ values for a single term t for each of the contexts in M corresponding to the audio-derived feature frames used to create $\mathbf{S}$. We create a $\mathbf{c}$ for each term in t.

To evaluate new data in a vector $\mathbf{x}$ (in a test set or on-line application), one simply computes the kernel product $K(\mathbf{S}, \mathbf{x})$ using the same σ as in training, and then multiplies the result by $\mathbf{c}_t$ for each t. The result will be a single scalar, which is usually regularized to its own sign $(-1 \ldots 1)$. This is the prediction of membership for audio frame $\mathbf{x}$ in term class t.

Scoring metrics

If the sign of $s(t, M)$ (or its thresholded version) is the same as our computed membership, we consider the prediction successful. The evaluation is then computed on the test set by computing a weighted precision: where $P(x_p)$ indicates overall positive accuracy (given an audio frame, the probability that a positive association to a term is predicted) and $P(x_n)$ indicates overall negative accuracy, $P(a)$ is defined as $P(x_p)P(x_n)$. However, to rigorously evaluate our term model's performance in a reaction prediction task, we note that this value has an undesirable dependence on the prior probability of each label and rewards term classifiers with a very high natural f_d, often by chance. For example, the term 'sad' could have a prior positive probability of P(sad) = 0.5 (the amount of frames overall that have a 'sad' ground truth.) If we guessed randomy with knowledge of the prior, $P(x_p)$ would be 0.5 and $P(x_n)$ would also be 0.5 ($P(x_n) = 1 - P(x_p)$.) This would make $P(a)_{sad}$ have a baseline of 0.25, 25%. We counter this by noting that none of our terms have priors close to 0.5, most have a f_d of under 0.1, which would make the baseline $P(a) = 0.1 \times 0.9 = 0.09$ (9%).

In practice, the $P(a)$ measure is a useful evaluation metric for term attachment. When the classifier accuracy needs to be measured in a more robust way, we use a model of relative entropy, using the Kullback-Leibler (K-L) distance to a random-guess probability distribution.

We use the K-L distance in a two-class problem described by the four trial counts in a confusion matrix, where *t* is 'sad' for the classifier $\mathbf{c}_{sad}$:

	Classifier says 'sad'	**Classifier says 'not sad'**
Ground truth says 'sad'	a	b
Ground truth says 'not sad'	c	d

a indicates the number of frames in which a term classifier positively agrees with the truth value (both classifier and truth say a frame is 'sad,' for example). b indicates the number of frames in which the term classifier indicates a negative term association but the truth value indicates a positive association (the classifier says a frame is not 'sad,' but truth says it is). The value c is the amount of frames the term classifier predicts a positive association but the truth is negative, and the value of d is the amount of frames the term classifier and truth agree to be a negative association. We wish to maximize a and d as correct classifications; by contrast, random guessing by the classifier would give the same ratio of classifier labels regardless of ground truth i.e. $a/b \approx c/d$. With $N = a + b + c + d$, the K-L distance between the observed distribution and such random guessing is:

Term	Precision		Parameter	Precision
busy	42.2%		big - little	30.3%
steady	41.5%		present - past	29.3%
funky	39.2%		unusual - familiar	28.7%
intense	38.4%		low - high	27.0%
acoustic	36.6%		male - female	22.3%
african	35.3%		hard - soft	21.9%
melodic	27.8%		loud - soft	19.8%
romantic	23.1%		smooth - rough	14.6%
slow	21.6%		clean - dirty	14.0%
wild	25.5%		vocal - instrumental	10.5%
young	17.5%		major - minor	10.2%

Table 6.1: On the left: select adjective terms discovered by the time-aware adjective grounding system. Overall, the attached term list is more musical due to the increased time-aware information in the representation. On the right: select automatically discovered parameter spaces and their weighted precision – the most semantically significant description spaces for music understanding uncovered autonomously by our system.

$$\begin{aligned} KL = {} & \frac{a}{N}\log\left(\frac{N\,a}{(a+b)\,(a+c)}\right) + \frac{b}{N}\log\left(\frac{N\,b}{(a+b)\,(b+d)}\right) \\ & + \frac{c}{N}\log\left(\frac{N\,c}{(a+c)\,(c+d)}\right) + \frac{d}{N}\log\left(\frac{N\,d}{(b+d)\,(c+d)}\right) \end{aligned} \tag{6.1}$$

This measures the distance of the classifier away from a degenerate distribution; we note that it is also the mutual information (in bits, if the logs are taken in base 2) between the classifier outputs and the ground truth labels they attempt to predict.

The results for a select set of terms using the $P(a)$ measure are shown on the left of Table 6.1. While the overall accuracy is low, we should consider the extremely low baseline of the problem itself compounded with our low trust in the ground truth used for this evaluation. We see immediately that more musically relevant terms are predicted with far higher accuracy. In this manner, we can easily remove low-scoring classes, both for data reduction and for accuracy. This type of evaluation provides keen insights into the amount of descriptive power certain terms have against acoustic content.

We can also use these results to visualize the spectral fingerprints of various descriptions. We perform the QBD task and then take the mean of all spectral content described as certain high-scoring terms, weighting each frame by its $s(t, M)$ salience score. Figure 6-1 shows two sets of comparisons. We see the expected result for 'quiet' versus 'loud' and a curious but understandable increase in the bass level bins of the 'funky' spectrum versus 'lonesome''s flat response.

foreign - native	foreign - domestic	dissonant - musical	physical - mental	partial - fair
empirical - theoretical	concrete - abstract	curved - straight	lean - rich	lean - fat

Table 6.2: Example synant relations.

6.1.2 Parameter Learning

Given a set of 'grounded' single terms, we now discuss our method for uncovering parameter spaces among them. This model assumes that certain knowledge is not inferred from sensory input or intrinsic knowledge but rather by querying a 'linguistic expert.' If we hear 'loud' audio and we hear 'quiet' audio, we would need to know that those terms are antonymially related before inferring the gradation space between them.

WordNet

WordNet [51] is a lexical database hand-coded by a team of lexicographers. Its main organization is the 'synset', a group of synonymous words that may replace each other in some linguistic context. The meaning of a synset is captured by its lexical relations, such as hyponymy, meronymy, or antonymy, to other synsets. A certain subset of adjectives in WordNet are organized in two polar clusters of synsets, with each adjective linking to some antonym adjective. The hypothesis is that descriptive relations are stored as polar gradation spaces, implying that we can't fully grasp 'loud' without also understanding 'quiet.' We use these antonymial relations to build up a new relation that encodes as much antonymial expressitivity as possible, which we describe below.

Synant Sets

We defined a set of lexical relations called *synants*, which consist of every antonym of a source term along with every antonym of each synonym and every synonym of each antonym. In effect, we recurse through WordNet's tree one extra level to uncover as many antonymial relations as possible. For example, quiet's anchor antonym is 'noisy,' but 'noisy' has other synonyms such as 'clangorous' and 'thundering.' By uncovering these second-order antonyms in the synant set, we hope to uncover as much gradation expressivity as possible. Some example synants are shown in Table 6.2.

The obvious downside of computing the synant set is that they can quickly lose their atonymial relation – following from the example above, we can go from 'quiet' to its synonym 'untroubled,' which leads to an synantonymial relation of 'infested.' We also expect problems due to our lack of sense tagging: 'quiet' to its fourth sense synonym 'restrained' to its antonym 'demonstrative,' for example, probably has little to do with sound. But with so many possible adjective descriptors and the large potential size of the synant set, we expect our connection-finding machines to do the hard work of throwing away the mistakes but looking to perception.

To create a set of grounded parameters, we simply search through the set of grounded single terms (as in the left side of Table 6.1) and average the $P(a)$ score for each polar

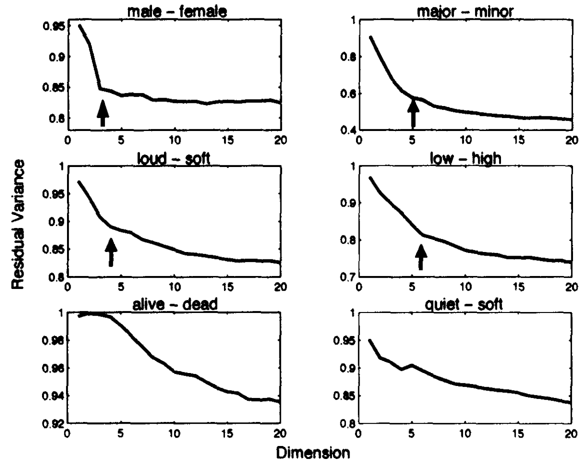


Figure 6-2: Residual variance elbows (marked by arrows) for different parameter spaces. Note the clear elbows for grounded parameter spaces, while less audio-derived spaces such as 'alive - dead' maintain a high variance throughout. Bad antonym relations such as 'quiet - soft' also have no inherent dimensionality.

side of all possible synants. For example, the $P(a)$ for 'quiet .. loud' is simply $P(a)_{quiet}$ + $P(a)_{loud}$ divided by two. This simple method has good results as shown in the right side of Table 6.1 – most of the groundable parameter spaces can all be considered musical.

Locally-Linear Embedding

We use the Isomap algorithm from [77] to attempt capture the structure of the audio features. Isomap scales dimensions given a $\ell \times \ell$ matrix of distances between every observation in ℓ. It roughly computes global geodesic distance by adding up a number of short 'neighbor hops' (where the number of neighbors is a tunable parameter, here we use $k = 20$) to get between two arbitrarily far points in input space. For our purposes, we use the same gaussian kernel (Equation 5.7) for a distance metric.

Isomap can embed in a set of dimensions beyond the target dimension to find the best fit. By studying the residual variance of each embedding, we can look for the elbow (the point at which the variance falls off to the minimum) – and treat that embedding as the 'innate' one. We use this variance to show that our highly-grounded parameter spaces can be embedded in fewer dimensions than ungrounded ones.

For each parameter space a_1 ... a_2, we take all observations automatically labeled by the test pass of RLSC as a_1 and all as a_2 and separate them from the rest of the observations. The observations F_{a_1} are concatenated together with F_{a_2} serially, and we choose an equal number of observations from both to eliminate bias. We take this subset of observation $F_{a_{12}}$ and embed it into a distance matrix D with the gaussian kernel. We feed D to Isomap and ask for a one-dimensional embedding of the space. The result is a weighting that we can give completely new unlabeled audio to and retrieve scalar values for each of these parameters.

By studying the residual variances of Isomap as in Figure 6-2, we can see that Isomap finds inherent dimensionality for our top grounded parameter spaces. We can look for the 'elbow' (either by sight or finding the maximum negative slope of the residual variance vector) which should define the dimensionality of embedding that maximizes information content. But for ungrounded parameters or non-antonymial spaces, there is less of a clear 'elbow' in the variances indicating a natural embedding. For example, we see from Figure 6-2 that the 'male - female' parameter (which we construe as gender of artist or vocalist) has a lower inherent dimensionality than the more complex 'low - high' parameter and is lower yet than the ungroundable (in audio) 'alive - dead.' These results allow us to evaluate our parameter discovery system (in which we show that groundable terms have clearer elbows) but also provide an interesting window into the nature of descriptions of perception.

6.1.3 Text Understanding

We next apply our work to the specific domain of record reviews. A system for review understanding is useful even to text-only retrieval systems: Consider a site that encourages on-line reviews of its stock; user-submitted text can be used in place of a sales-based collaborative filtering recommendation agent, and such systems prove to work well as buzz or opinion tracking models[21]. Of course, reviews have their problems. By their nature they are hardly objective – the author's own background and musical knowledge color each review. Music reviews can often be cluttered with outside-world information, such as personal relationships and celebrity trivia. While these non-musical tidbits are entertaining for the reader and sometimes (if obliquely) give a larger picture of the music in question, our current purpose would be best served by more concise reviews that concentrated on the contents of the album so that our models of music understanding and similarity are dealing with purely content-related features.

We chose 600 albums, two reviews for each (AMG [1] and Pitchfork [5]) to use later in interrater studies and as an agreement measure. Each pair $\{review, term\}$ retrieved is given the associated salience weight from the community metadata crawler. We limit the $\{review, term\}$ pairs to terms that occur in at least three reviews so that our machine learning task is not overwhelmed with negative bias. We perform the same query-by-description learning process as above in Section 6.1.1, with results using the K-L scoring metric in Table 6.3.

Many problems of non-musical text and opinion or personal terms get in the way of full review understanding. A similarity measure trained on the frequencies of terms in a user-submitted review would likely be tripped up by obviously biased statements like "This record is awful" or "My mother loves this album." We look to the success of our grounded term models for insights into the musicality of description and develop a review trimming system that summarizes reviews and retains only the most descriptive content. The trimmed reviews can then be fed into further textual understanding

adj Term	K-L bits	np Term	K-L bits
aggressive	0.0034	reverb	0.0064
softer	0.0030	the noise	0.0051
synthetic	0.0029	new wave	0.0039
punk	0.0024	elvis costello	0.0036
sleepy	0.0022	the mud	0.0032
funky	0.0020	his guitar	0.0029
noisy	0.0020	guitar bass and drums	0.0027
angular	0.0016	instrumentals	0.0021
acoustic	0.0015	melancholy	0.0020
romantic	0.0014	three chords	0.0019

Table 6.3: Selected top-performing models of adjective and noun phrase terms used to predict new reviews of music with their corresponding bits of information from the K-L distance measure.

Sentence	$g(s)$
The drums that kick in midway are also decidedly more similar to Air's previous work.	3.170%
But at first, it's all Beck: a harmonica solo, folky acoustic strumming, Beck's distinctive, marble-mouthed vocals, and tolls ringing in the background.	2.257%
But with lines such as, "We need to use envelope filters/ To say how we feel," the track is also an oddly beautiful lament.	2.186%
The beat, meanwhile, is cut from the exact same mold as The Virgin Suicides– from the dark, ambling pace all the way down to the angelic voices coalescing in the background.	1.361%
After listing off his feelings, the male computerized voice receives an abrupt retort from a female computerized voice: "Well, I really think you should quit smoking."	0.584%
I wouldn't say she was a lost cause, but my girlfriend needed a music doctor like I needed, well, a girlfriend.	0.449%
She's taken to the Pixies, and I've taken to, um, lots of sex.	0.304%
Needless to say, we became well acquainted with the album, which both of us were already fond of to begin with.	0.298%

Table 6.4: Selected sentences and their $g(s)$ in a review trimming experiment. From Pitchfork's review of Air's "10,000 Hz Legend."

systems or read directly by the listener. To trim a review we create a grounding sum term operated on a sentence s of word length n,

$$g(s) = \frac{\sum_{i=0}^{n} P(a^i)}{n} \tag{6.2}$$

where a perfectly grounded sentence (in which the predictive qualities of each term on new music has 100% precision) is 100%. This upper bound is virtually impossible in a grammatically correct sentence, and we usually see $g(s)$ of $\{0.1\% \text{ .. } 10\%\}$. The user sets a threshold and the system simply removes sentences under the threshold. See Table 6.4 for example sentences and their $g(s)$. We consider future work in this area of perceptual information retrieval: for example, the grounding sum operator could be used to only index certain terms in a webtext analysis and clustering agent for topic detection and similarity, enforcing that all the text is perceptually sound before the analysis.

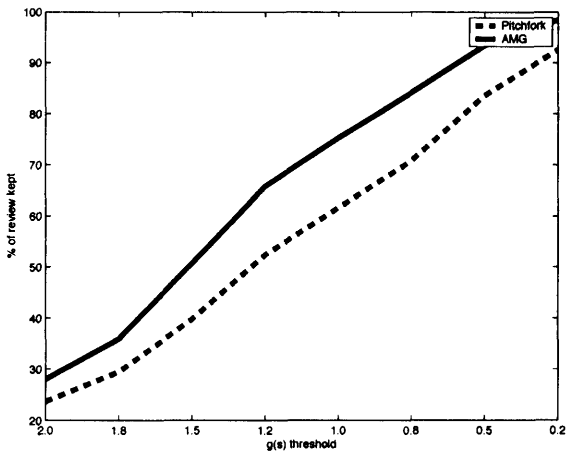


Figure 6-3: Review recall rates at different $g(s)$ thresholds for two different review collections. At lower $g(s)$ thresholds (shown towards the right of the x-axis) more of the review is kept in a summarization task.

We see that the rate of sentence recall (how much of the review is kept) varies widely between the two review sources; AMG's reviews have naturally more musical content. See Figure 6-3 for recall rates at different thresholds of $g(s)$.

6.1.4 Predicting Context from Content

To evaluate our perceptual link to audio in the context of reviews, we created a 'review prediction' task in which a new piece of audio is linked to its corresponding review. The evaluation is simply an accuracy measure: the number of albums whose audio was correctly linked with the correct review. We simply compute the term prediction task as above on half the albums of the record review set. For the second half, we generate our 'automatic review–' and if the terms lit up by the prediction have the closest match to the actual review, we consider it a successful prediction.

Using the same RLSC approach and features in Section 6.1.3, we chose 232 albums total, 116 albums per set. After computing the predicted review (a set of positive and negative term correlation for the audio) we compute Euclidean distance between the term vector and the truth term vector. We note this measure is not a very useful text similarity metric as it does not take into account the probabilities of the terms. With this metric, no song was accurately predicted, but overall the task scored 2.5% over the baseline (random), computed by finding the top predicted 50% mark (means of the predicted rank of the ground truth) at rank 951 out of 2000. The mean KL divergence of the task (as above but substituting reviews for terms) is 0.00032.

This task acts as a 'sanity check' for the term prediction and later semantic basis function music understanding task. The low results should not be alarming; review prediction is a very similar task to term prediction but averaged over an entire context instead of single term classifiers. With thousands of terms for each context, it would be hard to imagine that every classifier has good or even passable accuracy, and when considered in aggregate the poorly performing classifiers (which are removed in the rank reduction scheme described below) outweigh the well performing ones. A future review prediction task would have to take into account properly groundable terms as a threshold before prediction.

6.2 Semantic Rank Reduction

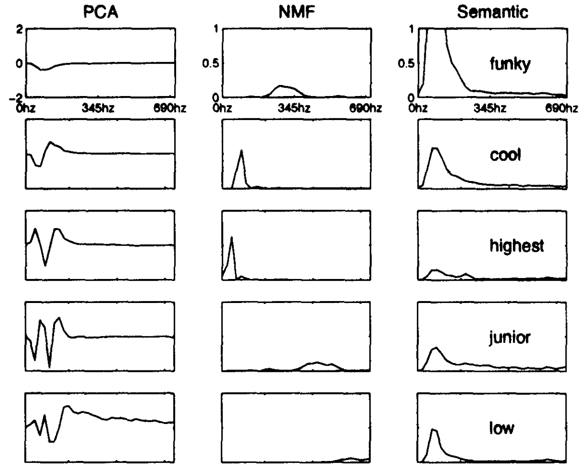


Figure 6-4: Comparison of the top five bases for three types of signal decomposition, trained from a set of five second power spectral density frames.

We use a method of 'semantic rank reduction' to test our hypothesis of meaning's importance to music understanding. As we've seen, using the outputs from our QBD classifiers we gain insights into the semantic attachment of various types of interpretation. In a closed set of music, we can list the r top-performing terms from an evaluation measure (either $P(a)$ or the K-L divergence) and then use those audio-to-interpretation classifiers on new audio. We can represent each frame of new audio as a combination of the top r term classifiers' outputs. In this manner we aim to connect statistical decorrelation techniques currently helpful in increasing the accuracy of music classifiers with the semantics of music, in effect creating a set of 'semantic basis functions' that can decompose new music audio into a compact representation that retains a maximal link from meaning to perception. We show that these $\mathbf{c}_t$ audio to term functions retain more information about the underlying music than other popular statistical decorrelation approaches evaluated in a music understanding task.

6.2.1 Anchor Models and 'Little Experts'

Our work in this area is based on research on anchor models: 'little experts' or combination classification. They are novel machine learning approaches with a clear vision:

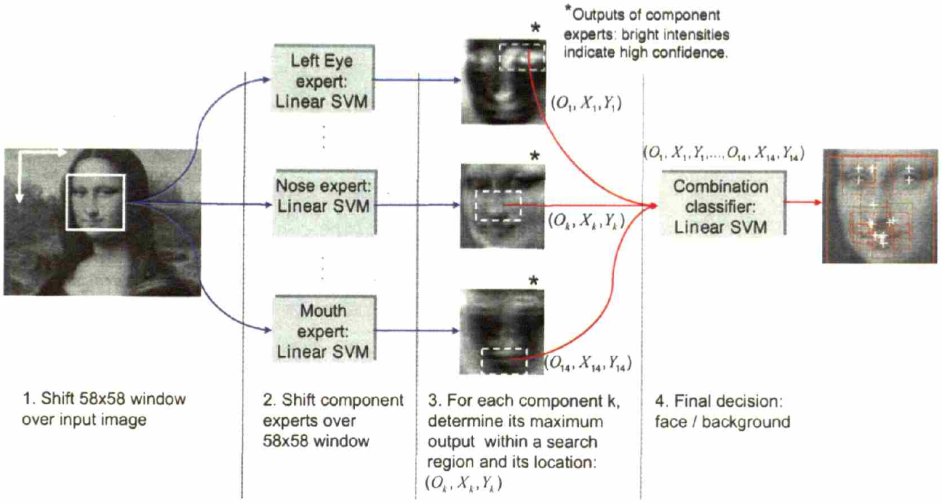
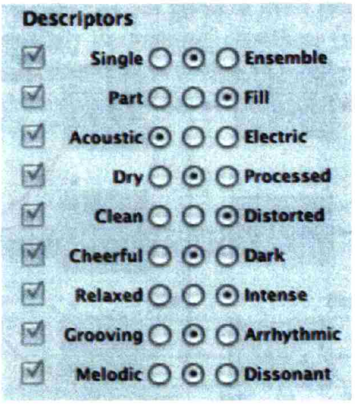

*Outputs of component experts: bright intensities indicate high confidence.
Left Eye expert: Linear SVM
Nose expert: Linear SVM
Mouth expert: Linear SVM
*
*
*
(O_1, X_1, Y_1)
(O_k, X_k, Y_k)
(O_{14}, X_{14}, Y_{14})
$(O_1, X_1, Y_1, \ldots, O_{14}, X_{14}, Y_{14})$
Combination classifier: Linear SVM
1. Shift 58x58 window over input image
2. Shift component experts over 58x58 window
3. For each component k, determine its maximum output within a search region and its location: (O_k, X_k, Y_k)
4. Final decision: face / background

Figure 6-5: Flow diagram of the Heisle et. al component combination face recognizer. [35]

instead of training a single machine to do a complex task, use the combination of a set of more refined experts to inform the larger machine. This allows for hierarchical levels of structure to be explicitly represented by the machine learning algorithm and process, or rather just as a simple human-readable way of 'getting' what is going on in the system. These processes usually mention a biological underpinning and have shown success in tasks that humans normally do well in, such as face recognition [35], where sliding windows moves over a face, looking for eyes, noses, or mouths. (See Figure 6-5 for a diagram.)

We note the difference between this approach and the 'mixture of experts' (ME) approach often used in neural network research [37] where the sub-classifiers are created and combined based on statistics of the task data. The above approaches have their sub-classifiers supervised (i.e. we have ground truth for each anchor model or little expert), where as in classic ME approaches, the task is 'semi-supervised' as only the larger task (the one whose error the sub-classifiers attempt to minimize) needs to have ground truth assigned. ME approaches also are not human-readable; their sub-classifier division is not based on any preconceived notion of the structure or semantic content of the underlying data, rather just statistics that cleanly separate the data. Our following work takes the 'supervision' out of anchor models while retaining this semantic link to the outside world.

In music, the notion of parameters or anchors is less clear and a current area of research. Some music applications such as Apple's Soundtrack (a composition program) tracks the semantics of its base component loops through a series of 10 descriptor pairs. (See Figure 6-6 for an example.) These anchors are manually entered. Recent work in music similarity and classification performed best with a set of 12 genre 'anchors' as



Descriptors

☑ Single ○ ● ○ Ensemble

☑ Part ○ ○ ● Fill

☑ Acoustic ● ○ ○ Electric

☑ Dry ○ ● ○ Processed

☑ Clean ○ ○ ● Distorted

☑ Cheerful ○ ● ○ Dark

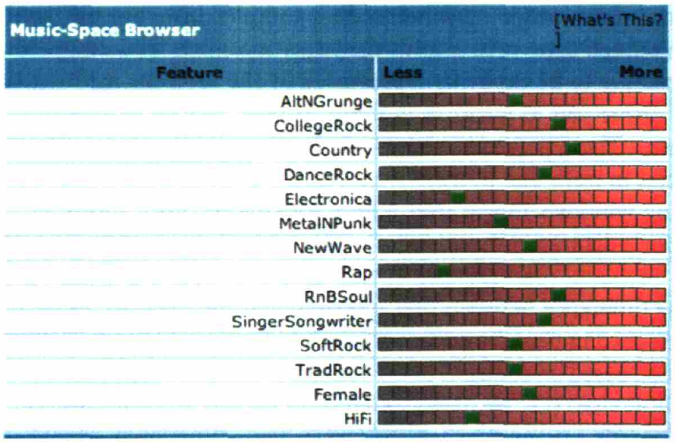
☑ Relaxed ○ ○ ● Intense

☑ Grooving ○ ● ○ Arrhythmic

☑ Melodic ○ ● ○ Dissonant

Figure 6-6: Musical anchors used in the Apple Soundtrack Loop Utility

well as two extra descriptors (male/female, and Lo-Fi/Hi-Fi) [12]. These anchors (see Figure 6-7) were manually entered in the training stage; once each sub-classifier was learned they can be evaluated on new data to create an anchored feature representation for later classification.



Music-Space Browser [What's This?]

Feature	Less More
AltNGrunge	
CollegeRock	
Country	
DanceRock	
Electronica	
MetalNPunk	
NewWave	
Rap	
RnBSoul	
SingerSongwriter	
SoftRock	
TradRock	
Female	
HiFi	

Figure 6-7: The genre+2 anchors chosen in Berenzweig et. al's anchor-space music similarity browser.[12]

In music, projecting data through a semantic anchor space is a tempting choice for its connection to outside-the-signal semantics. However, the main problem with the manual choice of anchors is that there is an immediate scientist bias: the experimenter chooses what they feel are the best descriptors for music. Even worse, in the above similarity case, the chosen anchors were mostly meaningless styles such as 'Singer/Songwriter' and 'New Wave.' We now note again our work above in parameter grounding, where, instead of manually determining important musical 'knobs,' we had the community define what they felt best separated large amounts of music. In this section we'll be showing how we created our own anchor space, defined more broadly as 'semantic basis functions:' and we'll show music projected through semantic basis functions to perform better at a very hard music understanding task than other rank reduction competitors that do not consider the meaning of music.

6.2.2 Obtaining a Semantic Decomposition

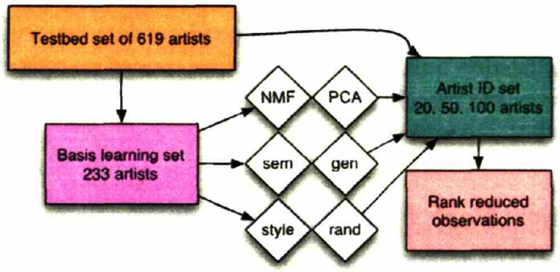


Figure 6-8: Process of the semantic rank reduction experiment. A large testbed set of audio is split into basis learning and artist ID sets. Various rank reduction methods (NMF, PCA, semantic, genres, styles and random) are performed on the artist ID set through the basis set and evaluated in a 1-in-n artist ID task.

We find our semantic decomposition by performing the query-by-description evaluation task outlined above, and keeping only the top r term classifiers, where r is the user's requested rank.

We choose a set of music audio, split it into equal-sized train and test segments, and label the observations with the artist names. The community metadata system retrieves the term types from the artist names creates community vectors for each artist a and term t. Concurrently, we form the audio from each artist into a frame-based representation. We then feed the training audio observations and the description vectors to a multiclass learning system to learn a new description vector for incoming audio frames.

We have the RLSC process create a $\mathbf{c_t}$ term classifier for each descriptor t in our crawl. To do so, we arrange a new $\mathbf{y_t}$ composed of the saliences for each descriptor on the fly. (For example, $\mathbf{y}_{sad}$ is a vector of the amount of 'sad' for each audio frame.) To determine which terms have stronger links between meaning and perception than others, we evaluate each $\mathbf{c_t}$ against the test set of audio using the $P(a)$ measure.

We sort the term list by $P(a)_t$, and leave it up to the user to select a rank r. The semantic basis functions are defined as the top r $\mathbf{c_t}$ classifiers ordered by our sort. (See Figure 6-4 for PSD bases of the top five classifiers kept in our experiment.) New data can be parameterized by a set of r coefficients, each one the result of asking the top audio-to-term classifiers to return a scalar of what they think of the incoming audio observation. This parameterization aims to retain maximal semantic value, where each dimension corresponds to some high-level descriptor of the input perception.

The set of $\mathbf{c_t}$ can be stored away for future use against any new set of music given that the representation of audio remains the same. For example, a generalized semantic rank reduction set of classifiers can be learned from a large set of all possible genres of music audio and later used against a new set of music audio. In this application case, the new set of audio does not need to be labeled with an artist tag or with description and we can view the semantic rank reduction of this data as an analogy to applying a weighting transform learned from a previous PCA (Section 6.2.4.) We note that some of the same caveats apply: bases should be learned from data that will be similar to data found in the classification task. Semantic classifiers trained on only classical music, for example, might retrieve specific term relations (such as 'bright' or 'brassy') and will not generalize well to rap music.

6.2.3 Statistical Rank Reduction Techniques

To evaluate our semantic rank reduction against other statistical techniques, we'll describe two currently popular methods of rank reduction used on audio signals, Principal components analysis (PCA) and non-negative matrix factorization (NMF.)

6.2.4 Principal components analysis

Principal components analysis (PCA) is a rank reduction technique that creates a weight matrix $\mathbf{w}$ and projection $\mathbf{f}$ of smaller rank than its source matrix $\mathbf{A}$ by extracting the principal components of $\mathbf{A}$. The principal components of a matrix $\mathbf{A}$ are the r eigenvectors of the covariance matrix $\mathbf{A}\mathbf{A}^T$ with the largest eigenvalues. The eigenvectors $\mathbf{w}$ of a matrix $\mathbf{A}$ when $\mathbf{A}\mathbf{w} = \lambda\mathbf{w}$. ($\lambda$ are the eigenvalues: λ is an eigenvalue if and only if $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$.)

We use the singular value decomposition (SVD) [32] to compute the eigenvectors and eigenvalues:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{6.3}$$

Here, if $\mathbf{A}$ is of size $m \times n$, $\mathbf{U}$ is the left singular matrix composed of the singular vectors of size $m \times n$, $\mathbf{V}$ is the right singular matrix matrix of size $n \times n$, and $\mathbf{\Sigma}$ is a diagonal matrix of the singular values σ_k. The highest singular value will be in the upper left of the diagonal matrix $\mathbf{\Sigma}$ and in descending order from the top-left. For the covariance matrix input of $\mathbf{A}\mathbf{A}^T$, $\mathbf{U}$ and $\mathbf{V}^T$ will be equivalent for the non-zero eigenvalued vectors. To reduce rank of the observation matrix $\mathbf{A}$ we simply choose the top r vectors of $\mathbf{U}$ and the top r singular values in $\mathbf{\Sigma}$.

To compute a weight matrix $\mathbf{w}$ from the decomposition we multiply our (cropped) eigenvectors by a scaled version of our (cropped) singular values: [74]

$$\mathbf{w} = \sqrt{\mathbf{\Sigma}^{-1}}\mathbf{U}^T \tag{6.4}$$

This w will now be of size $r \times m$. To project the original data (or new data) through the weight matrix one simply multiplies w by **A**, resulting in a whitened and rank reduced matrix f of size $r \times n$. To 'resynthesize' rank reduced matrices projected through **w** one first computes $\mathbf{w}^{-1}$ and then multiplies this new **iw** by **f**.

The intuition behind PCA is to reduce the dimensionality of an observation set; by ordering the eigenvectors needed to regenerate the matrix and 'trimming' only the top r, the experimenter can choose the rate of lossy compression. The compression is achieved through analysis of the correlated dimensions so that dimensions that move in the same direction are minimized. Geometrically, the SVD (and, by extension, PCA) is explained as the top r best rotations of the input data space so that variance along the dimensions is maximized.

6.2.5 NMF

Non-negative matrix factorization (NMF) [43] is a matrix decomposition that enforces a positivity constraint on the bases. Given a positive input matrix **V** of size $m \times n$, it is factorized into two matrices **W** of size $m \times r$ and **H** of size $r \times n$, where $r \leq m$. The error of $\langle \mathbf{W} \cdot \mathbf{H} \rangle \approx \mathbf{V}$ is minimized. The advantage of the NMF decomposition is that both **H** and **W** are non-negative, which is thought to force the decomposition to consider 'parts' in the observation space. Many applications of NMF have been proposed, including face analysis [42] and polyphonic music transcription [75]. The distance or divergence between **V** and $\langle \mathbf{W} \cdot \mathbf{H} \rangle$ can be measured by

$$D(V \| W \cdot H) = \| \mathbf{V} \times \log(\frac{\mathbf{V}}{\mathbf{W} \cdot \mathbf{H}}) - \mathbf{V} + \mathbf{W} \cdot \mathbf{H} \| \tag{6.5}$$

where × is a per-element multiply. The divergence measure here is found to be nonincreasing given the following two update rules:

$$\mathbf{H} = \mathbf{H} \times \frac{\mathbf{W}^T \cdot \frac{\mathbf{V}}{\mathbf{W} \cdot \mathbf{H}}}{\mathbf{W}^T \cdot \mathbf{1}} \tag{6.6}$$

$$\mathbf{W} = \mathbf{W} \times \frac{\frac{\mathbf{V}}{\mathbf{W} \cdot \mathbf{H}} \cdot \mathbf{H}^T}{\mathbf{1} \cdot \mathbf{H}^T} \tag{6.7}$$

where $\mathbf{1}$ is a $m \times n$ matrix of all 1.

6.2.6 Manual Anchors

Along with the statistical rank reduction methods, we also hand-chose two sets of 'manual anchors' in the form of Berenzweig et. al's work [12]. We chose a 'genre' set which contains seven genre terms and a 'style' set which contains up to 109 styles. The ground truth for the basis extraction step was pulled from the All Music Guide (AMG) [1] database.

Genre	Albums labeled	% Coverage	KL divergence
Alt-Rock	131	56%	0.0966
Electronic	67	29%	0.0988
Rock	20	8.5%	0.0202
Rap	9	3.9%	0.1104
Country	3	1.3%	0
Folk	2	0.8%	0.0193
Soundtrack	1	0.7%	0

Table 6.5: Ground truth coverage of genres in our manual anchor genre set along with KL divergence for each classifier in a genre prediction task.

Genre Anchors

The genre anchors were chosen directly from AMG over our test set of 233 albums. The coverage of this set is shown in Table 6.5. We see a wide bias for specific types of genres in our set, especially towards Alt-Rock and Electronic. Country, rap and folk are barely represented. This data is from a random subsampling of a large music database shared among students at MIT; it is meant to represent an 'average' music collection. The ground truth numbers alone show problems of coverage and scope.

Using these seven classes we predicted genres on a held out test set of music (1,000 songs fit into 10,000 observation frames) using the Penny feature; overall we achieved 69.3% per-song accuracy with a mean KL divergence of 0.0493 in predicting the correct genre. The per-classifier KL divergence measures are in Table 6.5. We note that the genres with the least coverage perform poorly, and that 'Rock' is not able to perform well but 'Alt-Rock' is.

Genre	Rock	Alt-Rock	Rap	Folk	Electronic
Rock	56%	6.5%	2.3%	0%	2.6%
Alt-Rock	36%	75%	4.6%	0%	25%
Rap	0%	1.2%	89%	0%	2.6%
Folk	0%	1.0%	0%	100%	0%
Electronic	8%	1.3%	4.6%	0%	68%

Table 6.6: Confusion for the 1-in-7 genre task (two genres had no data points.)

The genre identification confusion matrix in Table 6.6 shows that the task performed adequately overall (i.e. in a real world case, content labeled 'Rap' was guessed correctly 89% of the time) but there are some troubling confusions between the Rock and Alt-Rock tags as well as Electronic and Alt-Rock. (In our experiment test data, there was no randomly selected data labeled 'Country' or 'Soundtrack.')

Style Anchors

Table 6.7 shows coverage along *styles*, smaller clusters of music used for finer-grained classification. Again, we chose the ground truth styles from AMG for our test set. One major difference between genres and styles is that styles overlap and have a many-to-one relationship with artists and albums, a single album can have more than one style.

Style	Albums labeled	% Coverage	KL divergence
Experimental Techno	16	6.9%	0.0041
Singer-songwriter	19	8.2%	0.0200
Punk Revival	3	1.3%	0.0232
Shibuya-Kei	3	1.3%	0
Trip-Hop	20	8.6%	0.0017
IDM	28	12%	0.0040
Noise Pop	12	5.2%	0.0036

Table 6.7: Ground truth coverage of selected styles (from 109 total) in our manual anchor style set along with KL divergence for each selected classifier in a style prediction task.

As well, artists can be linked to multiple styles through multiple albums. Styles are a better solution to the 'Madonna problem' listed below for artist ID as they try to directly incorporate knowledge of trends over time. There naturally more styles than genres; in our set of 233 albums (1,000 songs, 10,000 frames) we had 7 distinct genres and 109 total styles. Due to the surfeit of styles we have even worse bias problems than genres, however; many styles have only one or two positive examples.

Overall, we achieved a 48.1% accuracy on the same 1,000 songs with a mean KL divergence of 0.0055 in predicting at least one of the ground truth styles (due to overlap in style labels, the KL measure is far more enlightening.) Only 64 of the 109 style classifiers had a KL divergence over $\epsilon = 2^{-23}$. The best performing style classifiers according to the KL measure were Hip-hop, Indie Rock, Underground Rap, Alternative Pop-Rock and Indie Pop.

For both the style and genre manual semantic reduction tasks, we stored the sorted list of classifiers for each style or genre along with their corresponding performance in the KL measure.

6.2.7 Semantic Basis Classifiers

Descriptor	KL divergence	Descriptor	KL divergence
punk	0.01996	drunken	0.00002
electronic	0.01447	cheap	0.00044
romantic	0.01313	famous	0.00020
moody	0.01251	confident	0.00002

Table 6.8: Term prediction results for the semantic basis classifiers. On the left: the top scoring classifiers with their KL divergence. On the right: selected badly-scoring classifiers.

We computed semantic basis functions using the RLSC method described above. For this experiment, we chose a subset of our entire community metadata, concentrating only on the adjectives. For our 233 artist basis learning set, we found 993 distinct adjectives from the webtext crawl. We split the basis set into two and evaluated the classifiers on the test set to achieve the ordered list of top-performing classifiers. We scored 72% accuracy in the positive prediction task (we guessed positive ground truth for the 1,000 songs 72% of the time overall) but this metric is not valuable due to the bias and prior of each classifier. In Table 6.8 we show instead a few selected classifiers (good on the left, bad on the right) with their KL divergence measure. Like the previous examples for query-by-description we see the set of terms on the left are 'musical' while the terms on the right are not groundable.

For the purposes of this experiment, we save the sorted list of classifier performance on the test set for later rank reduction.

6.2.8 "Basis Functions"

As the manual and automatic semantic classifiers we extract cannot be used to resynthesize the observation, they are not explicitly 'basis functions.' Likewise, the ordered list of the top r classifiers does not represent the maximal variance among coefficients. For example, the top performing classifier might be 'quiet' while the second best performing classifier could be 'soft.' As both point to the same type of semantic connection, there would be little difference between the two classifiers and as a result the 'functions' they create would not do well at maximizing variance among the observation's dimensions. And in the genre case, 'Rock' and 'Alt-rock' both contain a large amount of confusion with each other. Future work is concentrating at determining the best *combination* of classifiers, rather than arbitrarily choosing the best performing ones.

6.2.9 Evaluation using Artist Identification

We use an artist identification problem to evaluate different dimensionality reduction methods. Artist ID [82] [11] is a well-defined problem with obvious ground truth and requires a representation and learning algorithm that can capture a high level of musical information. Artist ID problems are usually formed as multi-class problems with a high number of output classes; as a result they benefit from dimensionality reduction steps that reduce noise in the input space.

There are two known problems with artist ID: 'The Producer Effect" (aka "The Album Effect"): an artist ID classifier might learn the production (EQ, compression, sound field) rather than the actual music. As well, there is the "Madonna Problem:" artists change over time. 1985 Madonna is not 2004 Madonna, but she's still Madonna. Which features actually define an artist? Often the 'producer effect' is minimized by training across albums when available, but in practice most artists stay with a 'sound' throughout their careers. Training on remixes or live versions as well as studio recordings could help this effect but the accuracy of these processes has yet to be evaluated.

6.2.10 Evaluation 1: Statistical vs. Semantic Basis Functions

In our first experiment we compare statistical methods such as PCA and NMF against the automatic semantic basis functions pulled from webtext.

Testing:	**non**	**pca**	**nmf**	**sem**	**rand**
Per-class	31.6%	28.6%	43.8%	66.3%	25%
Per-observation	22.2%	24.6%	19.5%	67.1%	3.9%

Table 6.9: Results for 1-in-20 artist ID experiment. Per-class accuracy is the $P(a)$ measure for RLSC bias correction averaged over all class t. Per-observation accuracy is a more natural metric: for each observation, was the artist classifier correct?

To evaluate artist ID, we start with a set of artists' songs and split the songs into two subsets: half of the artists for the basis extraction and half of the artists for the 1-in-20 artist ID task. Each artist was represented by five songs worth of material chosen randomly (across albums if available.) We compute a feature vector space on the entire music set, here we use Penny (Section 3.2.3) at 5 Hz.

We initially choose $r = 10$, and compute the PCA on the basis extraction set. We store only the transform weight matrix $\mathbf{PCA_w}$. We also compute the NMF in the same manner (over 5,000 iterations) and store its $\mathbf{NMF_w}$.

For the semantic basis function extraction, we subdivide the basis extraction set into two smaller sets of artists (training and testing) since labels are tied at the artist level, and use the community to get the set of $P(a)_t$. After performing the RLSC step (with $\mathbf{C} = 10$ and $\sigma = 0.5$) we evaluate against the basis extraction test set and retain the top 10 from our sorted list of $\mathbf{c_t}$ classifiers.

We then apply the stored $\mathbf{PCA_w}$ and $\mathbf{NMF_w}$ to the set used for the artist ID task. Each process creates an observation matrix of $r = 10$. To obtain the semantic reductions, we evaluate each point in our artist set against the stored $\mathbf{c_t}$ (for all the 993 adjectives as well as the genre and style terms), returning $r = 10$ scalar values for each classifier. (Note that we do not need to label the artist ID dataset with description, either from the genre/style labels or from the semantic basis functions after learning the decompositions on other data.) We arrange these results and treat the results as a $r = 10$ observation matrix.

The results for this experiment are shown in Table 6.9 along with the baseline (random) results. Confusion matrices for each experiment are in Figure 6-9. We see overall very high accuracy in training across the board, with perhaps the NMF hurting the accuracy versus not having an reduced rank representation at all. For the test case, results widely vary. PCA shows a slight edge over no reduction in the per-observation metric while NMF appears to hurt accuracy. We believe the NMF step is not a good fit for noisy audio observations where data is specifically not harmonic and easily separable. However, the semantic rank reduction step appears to do a good job in clustering the observations into a low dimensionality. It far exceeds the accuracy of a PCA pre-processing step and proves to be better than not doing any rank-reduction at all. Clearly the semantic reduction is informing the artist classifier in considering meaningful spectral characteristics not obviously present from statistical analyses.

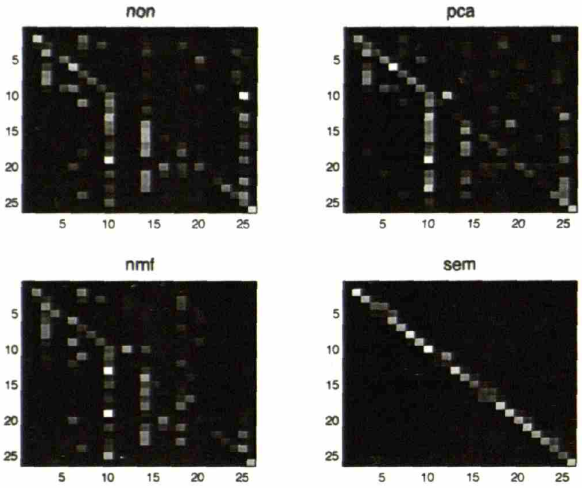


Figure 6-9: Confusion matrices for the artist ID experiments considering just the two statistical approaches (PCA and NMF) against automatic semantic rank reduction and no rank reduction. Lighter points indicate that the examples from artists on the x-axis were thought to be by artists on the y-axis.

Testing:	genre	style	sem	baseline
Accuracy	2.9%	4.7%	5.0%	1%

Table 6.10: Results for 1-in-100 artist ID experiment (song accuracy) comparing semantic basis functions with manual anchors, $r = 5$.

6.2.11 Evaluation 2: Semantic Basis Functions vs. Manual Anchors

Now that we've shown that statistical rank reduction does not perform as well as semantic rank reduction, we now look at two types of semantic reduction: the semantic basis functions vs. the 'manual anchors' discussed above. The genre and style anchors are computed as detailed above along with the semantic basis functions, using a gaussian kernel and auto-aux mode. We chose $r = 5$ since the genre classifier is limited to a $r = 7$. This is now a 1-in-100 artist ID task, a much harder task with a baseline of 1% over 1,000 songs.

We see in Table 6.10 that the style method outperforms the genres, but the semantic basis functions outperform both. The genre is especially a poor performer, and even when the same amount of classes are used for genres vs. styles ($r = 5$), the descriptive content of the style is greater.

6.2.12 Evaluation 3: Effect of Rank

We lastly vary the rank of the decomposition to see which type of rank reduction method worked better with less data. The point of this experiment is to find the

Accuracy:	genre	style	sem
$r = 1$	1.5%	1.9%	1.3%
$r = 2$	1.8 %	2.7%	1.5%
$r = 3$	2.9%	3.4%	1.9%
$r = 5$	2.9%	4.7%	5.0%
$r = 10$	NA	4.1%	7.7%

Table 6.11: Varying results for accuracy depending on requested rank during 1-in-100 artist ID experiment. Baseline is 1%.

amount of information towards the artist ID problem that each dimension along each reduction method provides. This sort of feedback is important for designing music understanding systems: the less dimensions required in the classifiers the faster they can execute with less memory.

Using the same approach above, we compute the manual and semantic anchors at varying r, from $r = 1$ to $r = 10$.

We can see in Table 6.11 that increasing the rank generally increases classifier accuracy across the board up to a certain 'elbow' point. Future work needs to consider the tradeoffs of rank in a semantic basis function, especially considering the overlap between different classifiers we discussed in Section 6.2.8. The semantic basis functions actually do worse with less rank than other manual anchors, but we have available to us up to tens of thousands of possible classifiers to choose among, while genre and style are limited to the dozens. Eventually the target dimensionality of the task (here, 100 for artist ID) should somehow inform the rank of the semantic reduction.

6.3 Conclusions

In this chapter we evaluate our meaning recognizer using two mechanisms: a prediction of interpretation from the audio, and through a music intelligence task (artist ID) after projecting the audio data through our semantic basis functions. We show that the meaning extraction is a necessary step for understanding music at a higher order than statistics can provide.

CHAPTER SEVEN
Conclusions

Throughout this thesis we've been re-iterating the idea that perhaps there's more to music than the signal, that perhaps looking at the content only is ill-advised and that looking at sales derived or marketing data is even worse. Our driving force behind this work is that fundamentally, the current approaches anger us: they don't seem right. Music is a personal force that resists 'processing,' 'packing' or 'understanding.' Our gasps at trying to approach music from two directions at once are illuminating but need to be refined and studied further. We're confident that looking at meaning is the right way to analyze music, but this feels to us like saying that looking at color is the right way to analyze images. Our hope is that this thesis can provide a framework for future music analysis systems to work within the realms of extra-signal and contextual data, and then linking that data to perception.

7.1 Our Meaning of Music

We can connect out work back to the meaning types we visited in Chapter 1 to get a handle on what we've accomplished and what is left to do. Our goal was to initially represent contextual and extra-signal data concerning music in the form of Community Metadata; this alone informs all three of our meanings: song stories and musical message in the correspondence meaning, similar artist description and artist clusters in relational meaning, and personal and cultural reaction as reviews and discussion. However, the main contribution of this thesis was to link and predict this meaning given a new signal, and in this model we were successful mostly at representing reaction meaning, both personal and cultural.

in Figure 7-1 we connect our approach with the meaning types, both of which we initially saw in Chapter 1. Our work in a contextual representation of audio allows us to 'understand' music at a different level than was previously possible. However, the work in prediction of reaction in semantic basis functions will inform many more audio understanding tasks as well as other multimedia tasks. The notion of grounding perceptual signals in interpretation is a growing field worthy of further study, and we present this work as the first step in music grounding.

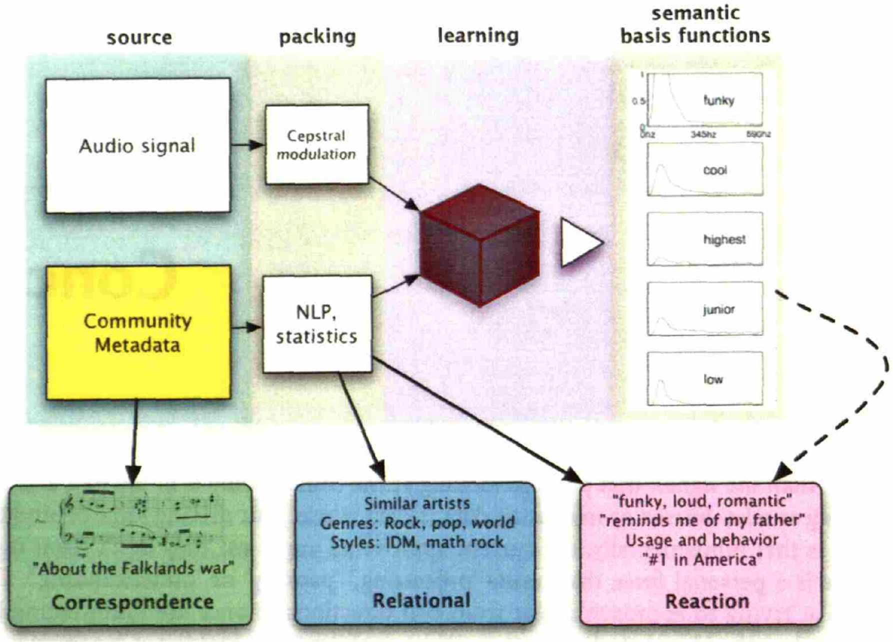


Figure 7-1: Linking our approach with types of represented meaning. Dotted lines indicate predictive extraction that can be applied to new (unheard) audio. Solid lines do not need an audio relation.

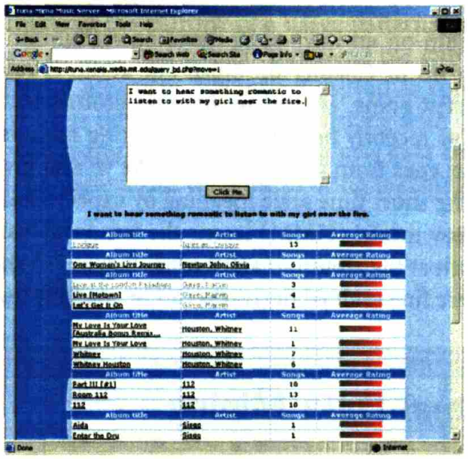
7.2 Query by Description as Interface

We note that we used query-by-description (QBD) as an evaluation task in Chapter 6, but never evaluated query-by-description as an interface, which is its first natural application. We found in small user tests that QBD, based on the audio and trained with the unsupervised term collection methods outlined in Chapter 5, was not effective for a general music search problem. We blame the sparsity of the term ground truth, the non-restraint on choosing the term ground truth, and the issues of bias and coverage outlined in Chapter 6.

A larger question is if QBD is valuable at all for music: there's a number of user interface and retrieval issues that need to be studied. Listeners are not used to describing types of music they want to hear (in a formal search-box type setting.) It's obvious that other external cues would be needed to make it valuable to users, such as drop down lists of artists or styles to choose among, or perhaps a user model that contains past listening experience and the transformations between query and result. It's also obvious that some terms with a high semantic attachment to audio via our analysis are not always understandable by human raters: we witnessed an ever-present 'junior' tag on top of most of our QBD prediction results. Although the tag is referring to something (we can plot its audio response) the returned music automatically tagged as 'junior' seems to have no graspable correlation. This could be a fault of either our feature extraction or learning system, but more likely the fault lies in our evaluation: what we do to determine which classifiers are the most *meaningful.*

An obvious step of future work is to perform a serious user study using QBD as the sole interface to interacting with music. We begun work on such an evaluation on a large music server where over 300 members of a community stream music to their players during the work day. Our plan is to replace the normal methods of browsing and searching with text-only and audio-linked QBD for certain users, and track relative success of relevance.

7.2.1 Text-only Query by Description


Music Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Google Search Web Search Site Page Info Up
Address http://ruva.xenakis.media.mit.edu/query_bd.php?move=1
Go
Click Me
Internet

I want to hear something romantic to listen to with my girl near the fire.

I want to hear something romantic to listen to with my girl near the fire.

Album title	Artist	Songs	Average Rating
[illegible]	[illegible]	13	▬
Album title	Artist	Songs	Average Rating
One Woman's Live Journey	Newton John, Olivia	6	▬
Album title	Artist	Songs	Average Rating
Live at the London Palladium	Gaye, Marvin	3	▬
Live [Motown]	Gaye, Marvin	4	▬
Let's Get It On	Gaye, Marvin	1	▬
Album title	Artist	Songs	Average Rating
My Love Is Your Love [Australia Bonus Remix...	Houston, Whitney	11	▬
My Love Is Your Love	Houston, Whitney	1	▬
Whitney	Houston, Whitney	7	▬
Whitney Houston	Houston, Whitney	6	▬
Album title	Artist	Songs	Average Rating
Part III [#1]	112	10	▬
Room 112	112	13	▬
112	112	10	▬
Album title	Artist	Songs	Average Rating
Aida	Sisqo	1	▬
Enter the Dru	Sisqo	1	▬

Figure 7-2: Query by description interface.

We do note that QBD was shown to work well in the *non-audio* case, that is, simply a similarity metric between query and webtext-community metadata (Figure 7-2.) This is a weaker approach as new audio as yet uncatalogued for context can not be evaluated or retrieved. However, for large collections we can assume coverage over much of the music, and we hope to work more in this text-retrieval only approach.

7.2.2 Description Synthesis

The presentable results we show in Figure 6-1 look promising but in reality there are only a dozen or so types of interpretation that can be appreciated in this manner. Even if a descriptor like 'angry folk' has a strong groundable connection to audio, what would it sound like? The inversion of the meaning classifiers is subject to much further work, hinging on a more expressive (and time aware) musical representation.

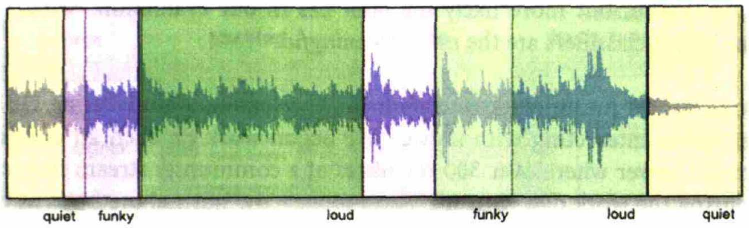
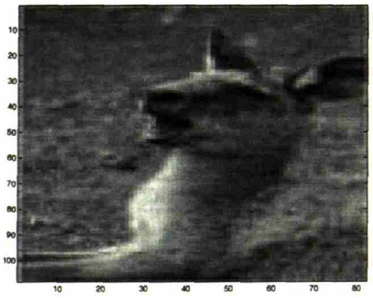


Figure 7-3: Example of meaning recognition output.

7.2.3 Time aware Meaning Recognition

Work has already begun on an intra-song meaning recognizer that addresses the issues of scale brought up in Chapter 6. Since our audio is related to only the level of ground context (artist name, album) we have no ground truth relating reaction to songs. However, we've recently been looking into song reviews to be able to train a 'bootstrap' model that goes from artists to songs. We've also investigated an unsupervised model of learning short time scale meaning by expectation-maximization (EM), where a hidden markov model (HMM) learns the temporal evolution of terms given the rough ground truth at the album or artist level. Figure 7-3 shows a possible output of such a system, where fine grained decisions can be made at the signal level.

7.3 Perceptual Data Mining



np Term	Score
austrailia exhibit	0.003
light and shadow	0.003
this incredibly beautiful country	0.002
sunsets	0.002
god's creations	0.002
the southeast portion	0.002
adj Term	**Score**
religious	1.4
human	0.36
simple	0.21
beautiful	0.13
free	0.10
small	0.33

Figure 7-4: Top terms for community metadata vectors associated with the image at left.

We want to close with a larger view on the problem: currently pattern recognition and media understanding systems optimize information as defined to a computer. Compression schemes pack bits in a way that can be efficiently and quickly unpacked by a machine later. Even perceptual coding schemes that acknowledge human interaction with the media rely on the bits containing all the information needed to get the message across. Following the analogy of information flow in Chapter 1, we should note that Shannon defines the informational content of a message X through the channel by its bits of entropy:

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2(p_i) \tag{7.1}$$

informed by the probabilities $p(i)$ of each symbol i in X. More 'surprising' symbols in a message need more bits to encode as they are less often seen. This equation commonly gives a upper bound for compression ratios and is often studied from an artistic standpoint. [54] In this model, the signal contains all the information: its significance is defined by its self-similarity and redundancy, a very absolutist view. Shannon information applied to a music signal reflects a particular type of meaning: applied to audio or the score, 'surprising' signals can affect the musical message as in correspondence meaning, and studying the entropy of a usage pattern can lead to understanding reaction in the form of buzz and trends. But alone the entropy of any signal can not cover our full forms of meaning. We intend instead to consider the meaning of those bits along with the bits themselves, and by working with other domains, different packing schemes, and methods for synthesizing new data from these significantly semantically-attached representations we hope to bring meaning back into the notion of information.

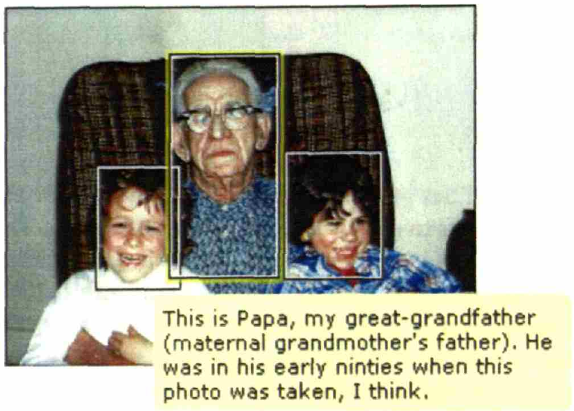
7.3.1 Images and Video

Low Term	Type	Accuracy	High Term	Type	Accuracy
antiquarian	adj	0%	sea	np	20%
boston	np	0%	pure	adj	18.7%
library	np	0%	pacific	adj	17.1%
analytical	adj	0%	cloudy	adj	17.1%
disclaimer	np	0%	air	np	17.1%
generation	np	0%	colorful	adj	11.1%

Table 7.1: Selected high- and low-scoring terms for an image description task.

Although our work has concentrated on music, we are very interested in other domains, especially images and video, as the models fit cleanly and have a more direct use. A query-by-description front end for image retrieval is a strong inspirational dream, and with the help of an already excited community, it appears the time is right to begin a large scale evaluation of the possibilities. In Table 7.1 and Figure 7-4 we point to our first try at performing the same meaning recognition on images using unstructured webtext. Our approach is similar to Barnard et al in [8], however, whereas they required hand-labeled single-term images, we would like to work from the free text surrounding an image.

The problems of image segmentation for object detection before meaning recognition is also in our reach with recent advances in online communities for photo sharing and distribution. The popular photo site Flickr [4] allows users to not only annotate their images with tags (single term descriptors, see Figure 7-6 for an example with cats) but also to physically segment the image in the web browser and annotate the segments as



This is Papa, my great-grandfather (maternal grandmother's father). He was in his early ninties when this photo was taken, I think.

Figure 7-5: Community-provided image segmentation and annotation. [4]

in Figure 7-5. This community-supplied data, en masse, is extremely useful for on-line meaning extraction systems that want to learn the identity and meaning of images and objects within images. It even could be more successful than the music experiments as the annotations are directly relating to a single image, not an entire body of work like most music description.

In the realm of video, recent work in video categorization and analysis interests us, such as the TREC Video Track, where text queries are given and, in a timed experiment, users must find the best matching video clips from a large database. These evaluations can be aided by taking advantage of contextual data such as viewer opinion and associated articles.

7.4 Fixing Music

Our work in this thesis is a first step toward adequately learning and representing the meaning of music, both by understanding the audio and its audience. As the fundamental problem of music distribution and organization sorts itself out over the next short while, we hope that these methods will be integrated into user interfaces, data mining systems and music similarity techniques as the other options are blind to the personal and cultural attachment to music. Specifically, we are interested in the problem of music recommendation – how to get people to find more music that they like – and the current culture- and meaning-blind methods are lacking in many ways. Through this work and its extensions we are looking forward to connecting music's interpretation to its perception, and the natural and musically knowledgeable applications should soon follow.

Figure 7-6: Community-provided image annotation: some of 32,000 'cat' images. [4]

Bibliography

[1] All music guide. http://www.allmusic.com.

[2] Art of the mix. http://www.artofthemix.com.

[3] BBC sold on song. http://www.bbc.co.uk/radio2/soldonsong/songlibrary/indepth/shipbuilding.shtml.

[4] Flickr. http://www.flickr.com.

[5] Pitchfork media. http://www.pitchforkmedia.com.

[6] E. Angerson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammarling, J. Demmel, C. Bischof, and D. Sorensen. LAPACK: A portable linear algebra library for high-performance computers. pages 2–11, 1990.

[7] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[8] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In *Proc. IEEE International Conference on Computer Vision*, pages 408–415, July 2001.

[9] M. A. Bartsch and G. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[10] A. Berenzweig, D. Ellis, B. Logan, and B. Whitman. A large scale evaluation of acoustic and subjective music similarity measures. In *Proc. International Symposium on Music Information Retrieval*, October 26–30 2003.

[11] A. Berenzweig, D. P. Ellis, and S. Lawrence. Using voice segments to improve artist classification of music. In *AES-22 Intl. Conf. on Virt., Synth., and Ent. Audio.* 2002.

[12] A. Berenzweig, D. P. Ellis, and S. Lawrence. Anchor space for classification and similarity measurement of music. In *Proceedings of ICME 2003.* 2003.

[13] E. Brill. A simple rule-based part-of-speech tagger. In *Proc. ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.

[14] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[15] W. Chai and B. Vercoe. Folk music classification using hidden markov models. In *Proc. International Conference on Artificial Intelligence*, 2001.

[16] W. Chai and B. Vercoe. Music thumbnailing via structural analysis. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

[17] W. W. Cohen and W. Fan. Web-collaborative filtering: recommending music by crawling the web. *WWW9 / Computer Networks*, 33(1-6):685–698, 2000.

[18] M. Cooper and J. Foote. Summarzing popular music via structural similarity analysis. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

[19] E. Costello. Shipbuilding, 1983.

[20] R. B. Dannenberg, B. Thom, and D. Watson. A machine learning approach to musical style recognition. In *In Proc. 1997 International Computer Music Conference*, pages 344–347. International Computer Music Association., 1997.

[21] K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *International World Wide Web Conference*, pages 519–528, Budapest, Hungary, May 20–24 2003.

[22] P. Duygulu, K. Barnard, J. D. Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proc. European Conf. Computer Vision*, 2002.

[23] D. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proc. International Symposium on Music Information Retrieval ISMIR-2002*, 2002.

[24] D. Evans and J. Klavans. Document processing with LinkIT. In *RIAO 2000*, 2000.

[25] D. A. Evans and C. Zhai. Noun-phrase analysis in unrestricted text for information retrieval. In *Meeting of the Association for Computational Linguistics*, pages 17–24, 1996.

[26] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advanced In Computational Mathematics*, 13(1):1–50, 2000.

[27] G. Flake. Nodelib.

[28] G. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, Boston, MA, August 20–23 2000.

[29] J. Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II, Proc. SPIE*, pages 138–147, 1997.

[30] G. Fung and O. L. Mangasarian. Proximal support vector classifiers. In Provost and Srikant, editors, *Proc. Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 77–86. ACM, 2001.

[31] D. Gibson, J. M. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *UK Conference on Hypertext*, pages 225–234, 1998.

[32] G. Golub and C. V. Loan. *Matrix Computations.* Johns Hopkins University Press, 1993.

[33] M. Goto. A chorus-section detecting method for musical audio signals. In *ICASSP 2003 Proceedings*, pages V–437–440, 2003.

[34] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[35] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *Neural Information Processing Systems (NIPS)*, 2001.

[36] L. Hofmann-Engl. Towards a cognitive model of melodic similarity. In *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval*, pages 143–151, Bloomington, Indiana, 2001.

[37] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[38] T. Jehan. Perceptual segment clustering for music description and time-axis redundancy cancellation. In *Proceedings of the 2004 International Symposium on Music Information Retrieval*, 2004.

[39] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[40] A. Klapuri, T. Virtanen, A. Eronen, and J. Seppänen. Automatic transcription of musical recordings. In *Proc. CRAC-2001 workshop*, 2001.

[41] J. Lafferty and G. Lebanon. Information diffusion kernels. In *Neural Information Processing Systems (NIPS)*, 2001.

[42] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, (401):788–791, 1999.

[43] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[44] D. B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[45] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proc. International Symposium on Music Information Retrieval.* ISMIR, October 23-25 2000.

[46] B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. Technical report, HP Labs.

[47] K. D. Martin. *Sound-source recognition: A theory and computational model.* PhD thesis, Massachusetts Institute of Technology, 1999.

[48] K. McKeown, J. Klavens, V. Hatzivassiloglou, R. Barzilay, and E. Eskin. Towards multidocument summarization by reformulation: Progress and prospects. In *Proc. 17th Nat. Conf. on Artif. Intel. AAAI-99*, pages 453–460, 1999.

[49] E. Meuller. *Natural language processing with ThoughtTreasure.* Signiform, 1998.

[50] L. B. Meyer. *Emotion and Meaning in Music.* University of Chicago Press, 1956. 307p.

[51] G. Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312, 1990.

[52] G. A. Miller. Five papers on wordnet. *Technical Report, Princeton University*, 1993.

[53] G. W. Peter Knees, Elias Pampalk. Artist classification with web-based data. In *Proceedings of the 2004 International Symposium on Music Information Retrieval*, 2004.

[54] J. R. Pierce. *Symbols, signals and noise - The nature and process of communication.* Harper & Row, New York, NY, 1961.

[55] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin dags for multiclass classification. In S. Solla, T. Leen, and K.-R. Mueller, editors, *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.

[56] H. Putnam. *Representation and Reality.* MIT Press, 1987.

[57] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.

[58] L. Ramshaw and M. Marcus. Text chunking using transformation-based learning. In D. Yarovsky and K. Church, editors, *Proc. Third Workshop on Very Large Corpora*, pages 82–94, Somerset, New Jersey, 1995. Association for Computational Linguistics.

[59] B. Recht and B. Whitman. Musically expressive sound textures from generalized audio. In *Proc. DAFX-2003*, 2003.

[60] T. Regier. *The human semantic potential.* MIT Press, Cambridge, MA, 1996.

[61] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[62] R. M. Rifkin. *Everything Old Is New Again: A Fresh Look at Historical Approaches to Machine Learning.* PhD thesis, Massachusetts Institute of Technology, 2002.

[63] D. Roy. *Learning Words from Sights and Sounds: A Computational Model.* PhD thesis, Massachusetts Institute of Technology, 1999.

[64] D. K. Roy. Grounding language in the world: Schema theory meets semiotics. *Special Issue of Artificial Intelligence Journal: Connecting Language to the World*, 2005.

[65] S. Rüping. Support vector machines and learning about time. In *IEEE International Conference on Accoustics, Speech and Signal Processing (ICASSP'03)*, 2003.

[66] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw Hill, 1983.

[67] E. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, (50):588–601, 1998.

[68] E. Scheirer. *Music-Listening Systems.* PhD thesis, Massachusetts Institute of Technology, 2000.

[69] C. E. Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:379–423, 623–656, 1948.

[70] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.

[71] P. Singh. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access.*, 2002.

[72] J. Siskind. Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic. *Artificial Intelligence Review*, 15:31–90, 2001.

[73] M. Slaney. Semantic-audio retrieval. In *Proc. 2002 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2002.

[74] P. Smaragdis. *Redundancy Reduction for Computational Audition, a Unifying Approach.* PhD thesis, Massachusetts Institute of Technology, 2001.

[75] P. Smaragdis and J. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

[76] S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the meaurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, 8:185–190, 1937.

[77] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[78] G. Tzanetakis. *Manipulation, Analysis and Retrieval Systems for Audio Signals.* PhD thesis, Princeton University, 2001.

[79] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In *Proc. Int. Symposium on Music Inform. Retriev. (ISMIR)*, pages 205–210, 2001.

[80] V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.

[81] B. Whitman and D. Ellis. Automatic record reviews. In *Proceedings of the 2004 International Symposium on Music Information Retrieval*, 2004.

[82] B. Whitman, G. Flake, and S. Lawrence. Artist detection in music with minnowmatch. In *Proc. 2001 IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568. Falmouth, Massachusetts, September 10–12 2001.

[83] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proc. Int. Computer Music Conference 2002 (ICMC)*, pages 591–598, September 2002.

[84] B. Whitman and R. Rifkin. Musical query-by-description as a multi-class learning problem. In *Proc. IEEE Multimedia Signal Processing Conference (MMSP)*, December 2002.

[85] B. Whitman and P. Smaragdis. Combining musical and cultural features for intelligent style detection. In *Proc. Int. Symposium on Music Inform. Retriev. (ISMIR)*, pages 47–52, October 2002.

[86] C. Yang. Music database retrieval based on spectral similarity. In *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval*, pages 37–38, Bloomington, Indiana, 2001.