Automatic Music Annotation

Research Exam
Spring 2005

Douglas Turnbull
Department of Computer Science, UC San Diego
dturnbul@cs.ucsd.edu

May 25, 2005

Abstract

In the last ten years, computer-based systems have been developed to automatically classify music according to a high-level musical concept such as genre or instrumentation. These *automatic music annotation* systems are useful for the storage and retrieval of music from a large database of musical content. In general, a system begins by extracting features for each song. The labels and features for a set of labeled songs are used by a supervised learning algorithm to produce a classifier. This classifier can then be used to provide labels for unlabeled songs. In this paper, we examine commercial and academic approaches to musical annotation involving genre, instrumentation, rhythmic style, and emotion. We also describe various musical feature extraction techniques that have been developed for musical genre classification systems. Lastly, we suggest the use of latent variable models as an alternative to the supervised learning approach for music annotation. We describe the correspondence latent Dirichlet allocation (Corr-LDA) model, which has been used for image annotation, and discuss how this model might be adapted for music annotation.

1 Introduction

When a listener scans the radio in search of an agreeable station, a decision about whether to settle or to keep searching is made based on a small clip of audio. Immediately the listener can determine if the radio station is broadcasting human speech, music, or silence. In the case of music, he or she can usually understand some notion of genre, instrumentation, and tempo. In some cases, the listener can identify the artist and title of the song if it is similar (or identical) to songs previously heard.

The question is whether automatic methods can also be employed to deduce high-level information from audio content. Recently, a number of automatic annotation systems have been developed that attempt to classify samples of low-level audio into categories based on high-level concepts. We consider low-level audio to be any bitstream that represents sound. High-level musical concepts include genre, instrumentation, emotion, and tempo.

We will specifically discuss the analysis of audio in the context of music, though many of the techniques also apply to other audio domains such as human speech, animal vocalizations, environmental sounds and sound effects. The music domain is interesting in that the audio content is rich with information associated with both frequency (pitch, timbre) and time (note onset, tempo). In addition, there has been a long legacy of academic study of music by theoreticians, musicologists, and more recently, by members of the fast-growing music information retrieval (MIR) community[1]. The growth of the MIR community can be attributed to the proliferation of digital multimedia content that has been made available though the Internet. The goal of the community has been to find novel ways to store musical content in a large database so that it can be retrieved in an efficient and useful manner ([Foo99],[FD02],[Pac03]). Some of the main research topics include novel query methods such as query-by-humming [DBT03], segmentation algorithms, similarity metrics, and automatic annotation systems.

[1] For more information, see the proceedings of the International Symposium on Music Information Retrieval (www.ismir.net).

In this paper, we will compare a number of existing annotation systems. The term annotation includes both class labels found by classifying music and real-valued musical measurements found using regression. Classification tasks include systems which classify music based on objective concepts such as artist recognition [Kau01] or the instrumentation [HPD03] as well as subjective concepts such as genre [TC02] or emotional content [LOL03]. Regression tasks include perception-based measures such as musical energy [PZ04] and emotional intensity [YL04]. The term annotation is more general in that we do not limit a system to a preestablished set of class labeled or dependent variables. That is, an annotation can be any type of data that represents another type of data and is often referred to as *meta-data*. For music annotation, we might consider a set of words, such as a song review, as the annotation of a low-level representation of a song. (For example, reviews for thousands of songs can be found at AMG Allmusic. See www.allmusic.com.)

Annotation is generally broken into two subproblems: feature design and learning/modeling. Feature design involves developing a useful representation of a song from the low-level audio signal. The result is a low-dimensional feature vector $\mathbf{x}$. Feature design usually includes the extraction of features, the integration of the features over time, and reducing the dimensionality of the final feature vector. Extraction involves applying digital signal processing (DSP) techniques to short-time ($\sim$25ms) segments of low-level audio to produce short-time feature vectors. A series of short-time feature vectors is combined using a feature integration scheme. Often, this results in a high-dimensional representation of the song. In this case, either feature selection and/or standard dimensionality reduction techniques, such as Principal Components Analysis (PCA), can be used to reduce the dimension of the final feature vector.

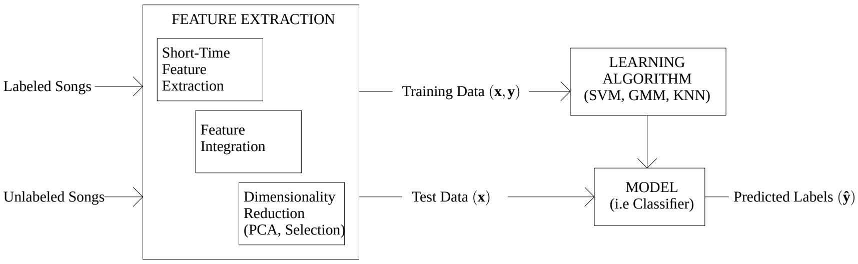
FEATURE EXTRACTION
Short-Time Feature Extraction
Feature Integration
Dimensionality Reduction (PCA, Selection)
Labeled Songs
Unlabeled Songs
Training Data $(\mathbf{x},\mathbf{y})$
Test Data $(\mathbf{x})$
LEARNING ALGORITHM (SVM, GMM, KNN)
MODEL (i.e Classifier)
Predicted Labels $(\hat{\mathbf{y}})$

Figure 1: Abstract learning problem for music annotation.

Existing systems approach annotation as a supervised learning problem. That is, a data set of $\{\mathbf{x},\mathbf{y}\}$ pairs, where $\mathbf{x}$ is a feature vector and $\mathbf{y}$ is an annotation vector is presented to a learning algorithm which outputs a model. (An annotation vector is often just a single class label.) During annotation, features for a novel piece of music, $\mathbf{x}_{new}$, are extracted and presented to the model which outputs an annotation vector $\hat{\mathbf{y}}_{new}$. A number of models, such as k-nearest neighbor (KNN) classifiers, mixture models, support vector machines (SVM), and artificial neural networks (NN), have been applied to music annotation.

One idea not found in the music annotation literature is the use of latent variable models. In a latent variable model, latent (i.e. hidden) variables are introduced which encode hidden states. Each state is used to model the joint distribution the content and the annotation. In Section 4, we develop this idea and show how a specific latent variable model, correspondence latent Dirichlet allocation (Corr-LDA), which has been shown to be a successful model for image annotation, might be adapted for music annotation. (Note that LDA, the acronym for linear discriminant analysis, is not associated with the Corr-LDA model.)

Although we will discuss human perception of sound and music in the context of feature sets that have been motivated by psychoacoustic models, we will not be focusing on the topic of how humans annotate music. While this is a very active area of research in computer music, psychoacoustic, and cognitive science communities, it is beyond the scope of this research paper. However, we refer the reader to the work of Lewicki [Lew02] which suggests, based on information theoretic principles, that humans use a combination of transforms resembling Fourier and wavelet transforms to encode natural sounds that are encountered in the world. The reader may also wish to refer to [Coo99] if interested in learning more about the physical nature of sound, the physiology of the human ear, and the field of psychoacoustics.

We begin by summarizing a number of commercial and academic systems that annotate music according to other high-level musical concepts including emotion, instrumentation, and tempo. We then provide a detailed review of four existing annotation systems that have been developed to classify music by genre.

We introduce the general concept of a latent variable model and describe the Corr-LDA model. We suggest that the Corr-LDA, and similar models based on this model, might be useful for music annotation. Our goal is to provide potential research directions for the music information retrieval community.

2 Musical annotation tasks

A review of existing commercial systems suggests that users are interested in searching for music by artists, title, year, genre, mood, instrumentation, tempo, and rhythmic style. In general, the commercial systems create large databases with human generated meta-data. Researchers have begun to develop systems that automatically classify music by a variety of high-level concepts. In this section, we will discuss both commercial and research systems that involve music annotation.

All the popular on-line (ITunes, Amazon, AMG AllMusic, MoodLogic) and off-line (ITunes, MusicMatch, Winamp) music management systems allow users to search by title, artist, album, and year. Annotation using these concepts is straight forward in that the information is encoded directly into the audio file using a tag. For example, an MP3 file will have ID3 tags which stores information about the artist, album, and year. If the tag is missing or corrupted, we can use automatic annotation to suggest or fill-in missing labels. While this task is not compelling from a commercial standpoint, it is a task, unlike genre, that has a ground truth and maybe useful for evaluating the relative performance of annotation systems.

Most music management systems also allow users to search for music according to genre/style. The term *style* is often used interchangeably with the term genre (i.e. see AMG AllMusic.) Some audio tags do allocate space for genre information, though neither standard hierarchy nor a method for determining the genre of a specific song exists. Rather, human determines both the structure of genre hierarchies and how to label each piece of music using ad hoc methods [PC00]. In Section 3, we review automatic systems that classifying music by genre.

Searching for music by emotional content has become popular in both commercial and academic systems. Moodlogic[2] uses collaborative filtering to collect high-level music labels from thousands of users for millions of songs. Users classify songs by genre, determine the predominant instruments, and rate songs according to both music concepts (energy, beat strength, danceability) and emotional concepts (happy/sad, mellow/not mellow, romantic/not romantic). Based on the collected content, users can search their personal music collection according to musical or emotional concepts. AMG AllMusic[3] allows a user to search a database of over five million songs on the basis of 179 moods (‘intense’, ‘lazy’, ‘spooky’) or by 85 themes (‘autumn’, ‘meditation’, ‘stay-in-bed’). The categories and song labels are determined by a large team of expert human reviewers. An expert is defined as someone who is professionally trained to annotate music.

Using automatic approaches to determine emotional content for unlabeled audio tracks, much like automatic genre classification, is an emerging topic. Li and Oghihara [LO03] label a data set of 499 songs according to sets of adjectives where each set represents a common emotion such as happy, depressing, or passionate. The features proposed in TC02 (see Section 3.1) are extracted for each of the songs and classification is performed using SVMs. Yang and Lee [YL04] automatically predict the *emotional intensity* of a song by formulating the task regression problem where emotional intensity is the dependent variable and low-level audio features (similar to the features described in [TC02]) are the independent variables. Emotional intensity, a real-valued measurement, is calculated based on how human listeners rate songs according a series of *single emotion tests*. In a single emotion test, a listener is asked to rate a song as according to bipolar pair of adjectives such as happy/sad or mellow/excited. More recently, Pachet [PZ04] uses a similar setup to predict *music energy*, another subjective measure related musical intensity.

Instrumentation is another concept which listeners use to search for music. The presence of an instrument or set of instruments (string, woodwinds, percussion) as well as information about the featured instruments (solo instrument, duration of the solo) is information that is useful to listeners, and especially musicians. However, relatively few music management systems allow query-by-instrument. One exception is AMG Allmusic, where a user can search the database by instrument for artists, but does not allow a user to find a specific song or a segment of a song that contain a solo.

Systems that automatically classifying isolated (monophonic) musical sounds by instrument have been developed with relatively high accuracy. For a review of a number of these systems, the reader can refer

[2] http://www.moodlogic.com

[3] http://www.allmusic.com

to [HPD03]. However, the problem identifying multiple instruments in polyphonic sound is more difficult. One approach has been to isolate individual instruments using source separation techniques such as independent component analysis (ICA). This approach has only shown limited success for toy data sets (i.e. duets or trios with instruments that have very different acoustic characteristics) since it requires estimates for the pitch of each instrument. This approach is not scalable in that when the frequency ranges of the instruments overlap, performance decreases. A promising alternative is to use supervised learning to detect the presence of instruments. In [ERD05], Essid et Al, achieve over 90% accuracy when trying to identify a combination between one and four instruments (drums, double bass, piano, trumpet, and saxophone) on a large data set of jazz trio/quartet songs.

Lastly, information about beat and rhythm is important for music annotation. A number of techniques, such as the beat histogram introduced in [TC02], can be used to find beats-per-minute, beat strength, and other rhythmic information directly from the audio content. Supervised learning can be used to classify a novel music track into rhythmic categories such as marches, waltzes, etc. In [GD04], a supervised learning approached is used to classify eight styles of ballroom dance music based on feature vectors containing 71 rhythm and onset features. Onset features involve information about musical notes such as the average duration of a note or the average time between successive notes.

It should be noted that in this section, we have discussed a number of commercial applications for automatic annotation. While these, and yet-to-be-invented applications do provide some economic motivation for annotation research, the primary goal is to find new and interesting ways to extract features and model data. Another goal is to provide general techniques that may be applied to other domains. For example, techniques from text document classification [BJ03] or images annotation [CV05] have make an impact on other research areas.

3 Four Genre Classification Systems

Musical genre is perhaps the most common concept used to classify music. It is used by record producers to target audiences, by musicologists to study musical influences, and by consumers to sort their private music collections. However, the notion of genre is inherently subjective in that neither the hierarchy of genres we use, nor the placement of songs into specific genres is universally agreed upon. For example, in a comparison of three Internet music providers, Pachet and Cazaly [PC00] found drastic differences in the number of genres, the words used to describe a genre, and the organizational structure of the genre hierarchies.

Despite the inconsistencies caused by its subjective nature, the concept of genre has received much attention from the MIR community. In general, the *subjectivity problem* is bracketed so that the researchers can make progress with the annotation problem. In each of the works we review, the authors make different assumptions about genre. These assumption are reflected in the number of genres, the names of the genres, and the criteria for labeling a song belonging to a genre. Furthermore, copyright laws and bandwidth issues prevent authors from establishing a common database of songs. The reader should be warned that both the authors' assumptions about genre and the use of different music databases affect classification performance, and thus, make it difficult to directly compare results.

In the following subsection, we review four music genre classification systems that were developed between 2002 and 2005. These systems have been chosen because of their contributions to musical feature design. Although the authors use the extracted features for the task of classification by genre, the features reflect general audio characteristics and can be used for other musical annotation tasks [LO03].(For brevity, we will denote each system by the first letter of each of the authors' last name and by the last two digits of the year of the principle publication describing the system.)

A review of existing genre classification systems prior to 2002 is provided by Aucouturier and Pachet in [pachet02]. The authors explore the features and learning algorithms used by six research-based systems, including the system developed by Tzanetakis and Cook [TC02] which is the first system we review. This system, denoted [TC02], incorporates a superset of features previously purposed and, in addition, introduces a number of novel Fourier-based feature extraction techniques specifically designed for music. The second system, developed by Li, Ogihara, and Li [LOL03], uses wavelet-based feature extraction techniques and support vector machines to achieve superior performance using the same data set as in [TC02]. Our third work by Martin and Breebaart [MB03] introduces new features based on human perception and a new feature integration technique using a filterbank transform. Our final work by Meng, Ahrendt, and Larsen [MAL05] compares the feature integration work of [MB03] to their own scheme based on autoregression.

The following subsections will involve summarizing concepts from digital signal processing and machine learning. The reader may wish to refer the Appendix for a review of these concepts.

3.1 TC02 - Tzanetakis and Cook (2002)

Prior to [TC02], the majority of audio feature extraction research was focused on extracting features for speech recognition and music-speech discrimination. By both using these existing techniques and developing new techniques specifically design for music analysis, Tzanetakis and Cook created a software system, called MARSYAS, which extracts a comprehensive set of musical features from an audio sample. Their features are divided into three sets: *timbral texture features, rhythm content features, and pitch content features.*

According to American National Standards Institute (ANSI), pitch loudness, and timbre are subjective measures of sound. Pitch is defined as an attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from high to low. Loudness is the attribute of auditory sensation in terms of which sounds may be ordered on a scale extending from soft to loud. Lastly, timbre is defined as attributes of sound that allows one to distinguish two sounds that are equal in pitch, loudness, and subjective duration. When analyzing sound, it is common practice to associate these three subjective concepts with some measurable quantity. Pitch is usually estimated as the fundamental frequency of the audio sample. Loudness is related to both sound intensity, measured in decibels, and frequency. Timbre is reflected in the shape of the spectrum in the frequency domain.

To summarize the timbre, a number of spectral shape features are calculated for each short-time window of audio content using the short-time Fourier transform (STFT). That is, the audio content is broken into short segments, usually about 25 ms in duration. Each segment is multiplied by a window function and transformed into the frequency domain using a Fourier transform. In the frequency domain, the segment is represented by a histogram which plots frequency verses magnitude. This histogram of the segment, called a spectrum, is summarized by four statistics: Root Mean Squared (RMS) energy, spectral centroid, spectral rolloff, and spectal flux. Another statistic, the zero-crossing rate, is calculated directly from the time series rather than from the spectrum. This statistic reflects the number of times the signal alternates between positive and negative amplitude and is a measure of the noisiness of the audio signal.

The Mel-Frequency Cepstal Coefficients (MFCC) are computed for each segment [RJ93]. (See Appendix A.3.) Based on experimental results, the authors choose to include only the first five MFCCs. Feature integration is done by calculating mean and variance of each statistic for a series of the short-time segments. The last feature, low-energy, is defined as the percentage of short-time segments with less RMS energy than average. This feature is intended to help discriminate between music that has many silent segments versus music without silent segments. The resulting 19 dimensional feature vector is given by

Timbral Texture features	
[1-8]	Mean and variance of centroid, rolloff, flux, and zero-crossing rate
[9]	Low-Energy: percentage of windows with less RMS energy than average
[10-19]	Mean and variance of the second through sixth MFCCs.

Based on their intuition in designing the low-energy feature, which is related to the skew of the empirical distribution of RMS energy, it is surprising that the authors did not include more than just the first moments of the empirical distribution (i.e. skewness and kurtosis). While this would increased the dimension of the timbral feature vector, dimensionality reduction techniques could be used to retain a low-dimensional representation.

The *rhythm content features* capture the period and beat strength of music based on a *beat histogram.* A beat histogram, which is calculated on 30-second segment of music, is a histogram of beat strength versus beats per minute (BPM). From the histogram, six features are calculated.

Similarly, the *pitch content features* are based on the analysis of pitch histograms. An *unfolded* pitch histogram is a plot of pitch strength verses frequency. The frequencies are usually binned so as to corresponded to musical notes. For example, middle A, which has a pitch of 440 Hz, will be the sum of pitch strengths between 428 Hz and 453 Hz. A *folded* pitch histogram is created by first combining all bins for

Rhythm Content features	
[20]	Total Beat Strength: sum of all beat strengths
[21-22]	Relative strengths of two highest peaks: peak strength divided by total beat strength
[23-24]	Period of two highest peaks in BPM
[25]	Ratio of two peaks strength: strength of second peak divided by strength of first peak

a musical note (e.g. A at 110 hz, 220 Hz, 440Hz, 880 Hz, etc), and then rearranging the bins so that they are ordered according to the circle of fifth (e.g. A is adjacent to D and E). Intuitively, the folded histogram gives a much more concise representation of the tonal qualities of the musical sample than the unfolded histogram. Using both histogram representations, five features are extracted

Pitch Content features	
[26]	Amplitude of highest peak in the Folded Pitch Histogram
[27]	Period of maximum peak in the Unfolded Pitch Histogram
[28]	Period of maximum peak in the Folded Pitch Histogram
[29]	Interval of two highest peaks in the Folded Pitch Histogram
[30]	Total Pitch Strength: sum of pitch strength over all frequencies

After the 30-dimensional feature vectors are extracted for a large database of labeled songs, the labels and vectors are used by a supervised learning algorithm to produce a model (i.e. a classifer). In TC02, the authors use K-nearest neighbor(KNN) classifiers and Gaussian Mixture Models (GMM). Based on a database containing 100 30-second song samples from each of 10 genres, the authors are able to achieve 61% classification accuracy using GMM with 3 mixtures per genre. This is compared with 70% classification accuracy by non-expert humans in a forced-choice test[4]. In later work, using the same features and database, Li and Tzanetakis [LT03] achieved 71% accuracy using linear discriminant analysis (LDA), and similarly, Turnbull and Elkan [TE05] achieved 71% accuracy using radial basis function networks (RBFN).

3.2 LOL03 - Li, Ogihara, and Li (2003)

When we compute the Fourier transform of a music signal, we lose all temporal information, unless we break the signal up and compute the short-time Fourier transform (STFT). Thus, the temporal information is limited by the segment length. However, using shorter segments yields poorer frequency resolution. Since the Fourier transform is a transform that has fixed frequency resolution across all frequencies, temporal information at high frequencies is poorly represented. The wavelet transform allows for variable resolution across the frequency range. The result is a time-frequency representation of a signal that has good time resolution for high frequencies and good frequency resolution at lower frequencies.

In [LOL03], Li, Ogihara and Li apply Daubechies wavelet transforms [Dau92] to music samples. The specific Daubechies transform decomposes an signal into seven frequency bands using seven filters. The filters all have the same shape but the differ in length. (i.e. The longer the filter the lower frequency.) Each filter is repeatedly convolved with the music signal then shifted by the length of the filter. Each convolution produces coefficient are used to make a histogram. The result is seven histograms, each of which are converted to empirical probability distribution functions. The first three moments (mean, variance, and skewness) and the average sub-band energy, the mean of the absolute value of thw coefficients, are calculated. This results in a feature vector of $7*4 = 28$ features. Based on empirical results, the authors find that the features from 3 bands are not informative and are dropped. Finally, the 19 timbral texture features from TC02 are added to the feature vector. The resulting feature vector given by

Daubechies Wavelet Coefficient Histogram (DWCH) features	
[1-17]	Daubechies Wavelet Coefficient Histogram 4 sub-bands * (mean, variance, skewness, energy)
[17-35]	Timbral Texture Features from TC02

[4] A forced-choice test requires that individual pick one genre from the set of 10 given genres.

Using the DWCH set of features, the authors compare a number of models: Gaussian mixture models (GMM), linear discriminant analysis (LDA), k-nearest neighbor (KNN) models, and three variants of Support Vector Machines (SVM). The most successful model is a one-versus-all SVM, in which one binary classifier is trained for each class using the examples from that class as positive examples, and the rest of the data as the negative examples. Each SVM outputs a score s that is mapped to a posterior probability $P(y|s)$ according to a function $f(s) = P(y|s)$ that is learned from training data [ZE02]. The final predicted class of an unlabeled data point is given by the classifier that produces the highest posterior probability. Using the DWCH features with a one-versus-all SVM, the authors report an improvement of 6.5% in classification accuracy over the best results reported by LT03 using LDA for the same data set of music songs.

3.3 MB03 - Martin and Breebaart (2003)

In [MB03], McKinney and Breebaart introduce two sets of audio features based on perceptual models of the human auditory system. The first set, the psychoacoustic features (PA), models the musical percepts of roughness, loudness and sharpness. The second set, the auditory filterbank temporal envelopes (AFTE), involves passing the audio signal through a gammatone filterbank. A gammatone filterbank is a specific set of basis functions for a wavelet transform that approximates empirical non-linear frequency response of the cochlea in the human ear. In addition to the PA and AFTE feature sets, the authors consider two existing features sets: MFCC features, and standard low-level (SLL) [LSDM01]. The nine SLL features are statistics that characterize the spectrum of the audio spectrum and are similar to the timbral and pitch features developed in TC02. The four feature sets are

Standard low-level (SLL) features	
[1-9]	RMS, centroid, bandwidth, zero-crossing, roll-off, band energy ratio, delta spectrum magnitude, pitch, pitch strength

Mel-Frequency Cepstral Coefficients (MFCC) features	
[1-13]	MFCC coefficients

Psychoacoustic (PA) features	
[1-2]	Roughness (mean and standard deviation)
[3]	Loudness
[4]	Sharpness

Auditory filterbank temporal envelopes (AFTE) features	
[1-18]	Gammatone wavelet coefficients

The second contribution of MB03 is a feature integration technique using a filterbank transform to produce *dynamic* features. In previous systems, the final feature vector is given by the statistical moments (e.g. mean, variance, skewness, kurtosis) of each feature over a series of the short-time windows. (See TC02 and LOL03.) One drawback of this *statistical moments* integration technique is that the temporal information for the time series is lost. In MB03, dynamic features are found calculating the power spectrum for the series of each short-time feature. The power spectrum is broken into four bands: 1) 0 Hz which is the DC component and is the mean of the feature, 2) 1-2 Hz which is the frequency range for most musical beats, 3-15 Hz, and 20-43 Hz. The final feature vector then has four times the dimension of a short-time feature vector.

Using one database of music (188 songs, 7 genres) and one learning algorithm (quadratic discriminant analysis), the authors compare the four feature sets (SLL, MFCC, PA, AFTE), and for each, compare the effect of incorporating the dynamic features with the static (means only) features. Since the size of each feature set is different, the authors use forward stepwise selection to select the nine best features from each of the eight sets for features. The results show that using dynamic features improves classification accuracy and AFTE features produce the best results. An interesting experiment that was not reported would have been to compare feature sets without using feature selection. Another idea that was not explored experimentally, though was mentioned by the authors, is to use a combination of features from the each of the sets features. This would result in a high-dimensional representation that may require the use of PCA to reduce the dimensionally of the feature vectors before modeling.

3.4 MAL05 - Meng, Ahrendt, and Larsen (2005)

The work of Meng, Ahrendt and Larsen ([MAL05]) focuses on early and late *information fusion.* Early information fusion, which includes feature integration, combines short-time (30 ms) features in order to create medium-time (740 ms) and long-time (9.62 s) features. The authors compare three feature integration methods: calculating means and variances (MV) of features ([TC02], [LT03], [TE05]), using the dynamic features proposed in [MB03] and referred to as filterbank coefficients (FC), and a novel method based on autoregression (AR). The features for the AR model are the linear predictive coding (LPC) coefficients for a given model order. The model order is determined by minimizing the classification error on a validation set. When the feature integration method results in a high dimensional feature vector, PCA is used for dimensionality reduction.

Late information fusion combines the outputs from a classifier computed on each of the shorter time audio feature vectors. (That is, outputs from short-time classifier are used to make medium-time and long-time decisions, and outputs from medium-time classifiers are used to make long-time decisions.) The authors pick the genre of a song according to the *sum rule*: the genre with the largest sum of posterior probability is chosen. This late fusion scheme requires that the outputs from the shorter time classifiers are probabilistic (i.e. output posterior probabilities $P(y|\mathbf{x}.)$ If the outputs are not probabilistic, either the outputs can be converted to probabilities using calibration [ZE02], or a different late fusion scheme, such as *majority vote* can be used. In this scheme, each shorter time classifier casts one vote for a single genre and the genre with the most votes wins.

The authors compare the a number of early and late information fusion schemes for classifying music by genre. In all cases, the short-time features are the first six Mel-frequency Cepstral Coefficients (MFCC). Using these six features, they create medium-time and long-time feature vectors using combinations of MV, FC and AR feature integration techniques. For each set of feature vectors, a probabilistic model is trained and performance is measure on a test set. The author use two models, neural networks and Gaussian mixture model, though few details about these models are given. Classification performance is compared against late information fusion schemes. The best results are found by using a three step fusion scheme: 1) calculate MFCC features for each short-time series, 2) integrate the short-time MFCC features using AR to find medium-time feature vectors, and 3) use late fusion according to the sum rule over the series of medium-time features. On a small data set contain 100 songs and five genres, this fusion scheme produces a error rate of 5% which is compared with a 3% error rate by human test subjects. All other fusion schemes resulted in significantly worse error rates of 12% or more. Similar results were found when using a large data set containing 354 song and six genres.

4 Unsupervised Annotation

Recently there has been a considerable amount of research on automatic image annotation: providing a caption for a given image. A brief review is given by Carneiro and Vasconcelos [CV05]. Initial work posed the problem as a supervised learning problem in which features are extracted from the image and a classifier is trained to recognize individual semantic keywords. Keywords denote holistic concepts, such as ‘indoors’, ‘outdoors’, ‘cityscape’, and ‘landscape’, or objects that appear in images such as trees, buildings, and horses. This supervised approach is very similar to all the approaches that have been used for music annotation.

A second general approach to image annotation involves introducing a set of latent variables that encode hidden states. Each state represents a joint distribution between keywords and image features. During training, a set of labeled images, each defined by a content-based feature vector and a set of keywords, is presented to an unsupervised learning algorithm, such as Expectation Maximization (EM), in order to estimate the joint distribution between image features and keywords. During annotation, keywords for an

Table 1: Comparison of four music genre classification systems. The major contributions are reflected in the bold font. The brackets after the feature sets denote the dimension of the final feature vector. For example, LDD (9/36) represents the low-level descriptor set of 9 static or 36 dynamic features. Acronyms: LLD - Low-Level Descriptors (FFT-based), MFCC - Mel-Frequency Cepstral Coefficients, GMM - Gaussian Mixture Model, KNN - K-Nearest Neighbor, SVM - Support Vector Machine, LDA - Linear Discriminant Analysis, QDA - Quadratic Discriminant Analysis, RBFN - Radial Basis Function Networks, NN - Neural Network, GC - Gaussian Classifier with full covariance matrix

System	Extraction	Integration	Selection	Model
TC02	**Timbral Texture** **(LLD (9) & MFCC (10)) ,** **Rhythmic Content (6),** **Pitch Content (5)**	means, variances	all	GMM, KNN
LT03	”	”	”	SVM, LDA
TE05	”	”	”	RBFN
LOL03	**Daubechies Wavelet** **Coefficient Histogram (16),** Timbral Texture (LLD(9) & MFCC(10))	means, variances	all	SVM, LDA, KNN, GMM
MB03	LLD (9/36), MFCC(13/52), **Psychoacoustic, (3/10)** **Auditory Filterbank (18/62)**	means, **power spectrum**	**stepwise** **selection(9)**	QDA
MAL05	MFCC(6)	means, variances, power spectrum, **autoregression,** **late fusion**	all	NN, GC

Table 2: Reported Results of music genre classification systems. In general, results are based on each author's internal database of songs and assumptions about genre, which greatly effect performance. The exception (denoted by *) is the database created for TC02 which was used in TC02, LT03, TE05, and LOL03. The performance by non-expert human listeners for this data set is reported as 70% in TC02.

System	Task (# classes)	Database Size	Best Results
TC02	Genre(10)*	1000	61%
	Classical(4)	400	88%
	Jazz(6)	600	68%
LT03	Genre(10)*	1000	72%
TE05	Genre(10)*	1000	71%
LOL03	Genre(10)*	1000	79%
	Genre(5)	756	99%
MB03	Audio(5)	310	93%
	Genre(7)	188	74%
MAL05	Genre(5)	100	96%
	Genre(6)	354	69%

unlabeled image are the set of keywords that maximize this joint distribution. To the author's knowledge, little (or no) research has been focused on applying this unsupervised approach for music annotation.

4.1 Correspondence LDA for Music Annotation

One popular generative model for image annotation is correspondence latent Dirichlet allocation (Corr-LDA) introduced by Blei and Jordan in [BJ03]. The graphical model representation for Corr-LDA is given in figure 2. Consider an image as being represented by the $(\mathbf{r}, \mathbf{w})$ pair, where $\mathbf{r}$ is a vector of N region-specific feature vectors and $\mathbf{w}$ is set of M keywords. The N regions of the image can be found using block-based decomposition of the image (i.e. cutting the image up into rectangles) or using image segmentation techniques such as the N-cut algorithm. The vocabulary of keywords, denoted by W, is selected from a corpus of image captions.

The generative process for each image $(\mathbf{r}, \mathbf{w})$ under the Corr-LDA model is:

1) A random variable θ is drawn a from Dirichlet distribution with parameter α and defines a distribution over hidden states.

2) For each image region $\mathbf{r}_n$ in the image, a random variable z_n representing a hidden state is drawn

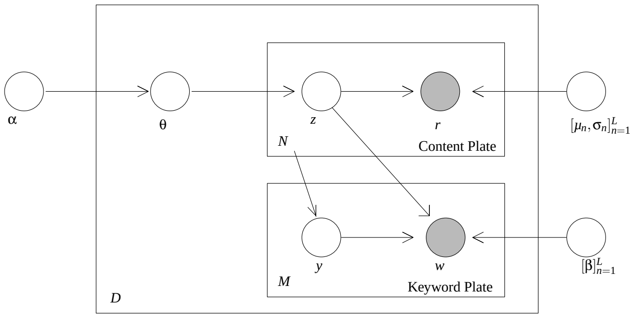

Figure 2: Corr-LDA Graphical Model

from a multinomial distribution with parameter θ. Then the feature vector $\mathbf{r}_n$ is drawn from a multivariate Gaussian distribution with parameters $\{\mu_n, \sigma_n\}$ associated with state z_n.

3)For each keyword w_m, a random variable y_m is drawn from uniform distribution with parameters (1,N). This random variable represents one of the hidden states, denoted z_{y_m}, that was used to generate an image regions in the step 2. A keyword w_m is drawn from a multinomial distribution with parameter β_n associated with the hidden state z_{y_m}.

Parameter estimation for the Corr-LDA model is done using variational Expectation Maximization (EM) [Ble04]. Annotation of an unlabeled image $\mathbf{r} = \{\mathbf{r}_1, ..., \mathbf{r}_N\}$ involves finding the M keywords that maximize the conditional distribution of the keywords given $\mathbf{r}$ and the model parameters. It is not clear in the paper exactly how annotation is performed, but we assume that the M keywords are the M keywords that individually maximize the conditional distribution. The equation for picking the M words is

$$\text{argmax}_{\mathbf{w} \in W} \prod_{i=0}^{L} \prod_{n=1}^{N} p(\mathbf{r_n} | \mu_i, \sigma_i) p(w | \beta_i) \tag{1}$$

where L is the number of latent states, W is the vocabulary, and $\mathbf{w}$ represents the M.

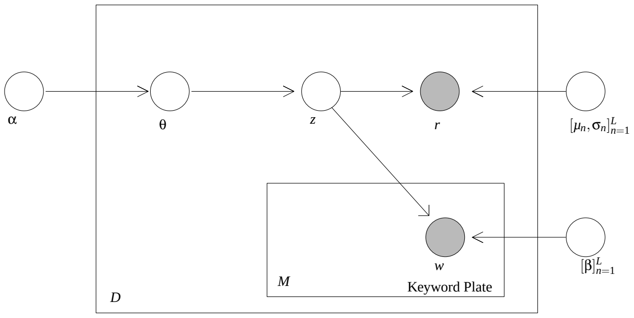

Figure 3: Corr-LDA Graphical Model for Music Annotation where each song is represented by one segments.

When applying applying the Corr-LDA model to music annotation, the observable feature vector x will be composed of musical features such as the features described in Section 3. The keywords $\mathbf{y}$ will come from a musical vocabulary and can be any word that is used to describe music, such as words related to genre, instrumentation, emotion, etc. If we consider each song as being composed as one segment, the Corr-LDA model is simplified. That is, only one random variable z_1 representing one hidden state will be used to generate the entire song and all of the keywords for the song. This graphical model is shown in Figure **??**. However, audio segmentation can be done by taking fixed length segments of audio or by using automatic music segmentation techniques [PE04]. If we allow for audio segmentation, the Corr-LDA model is the same as the model given in Figure 2. It is interesting to note that this model simultaneously relates information about a number of music concepts. This is a major advantage over existing systems which only classify based on one music concept.

One drawback of the Corr-LDA model is that annotation is not guaranteed to return words related to all of the various musical concepts. For example, the annotation of a song might produce many words related to genre but no words related to instrumentation. A simple technique to resolve this problem is, for each task, output the top M' words associated with that concept according to Equation 1. A second technique is to modify the graphic model to account for a hierarchial vocabulary. (See Figure 4.) This new graphical model, called hierarchical Corr-LDA, that can incorporate a hierarchical vocabulary is given in Figure 5. By breaking the vocabulary into smaller (possibly disjoint) sets, parameter estimation might be improved. Comparing these two techniques both in terms of performance and scalability is an interesting future research direction.

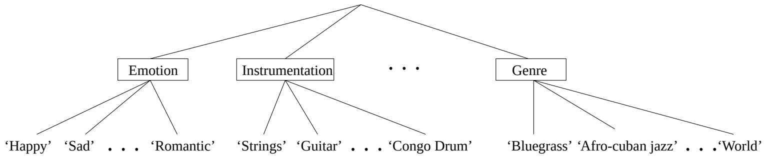

Figure 4: Hierarchical Music Vocabulary

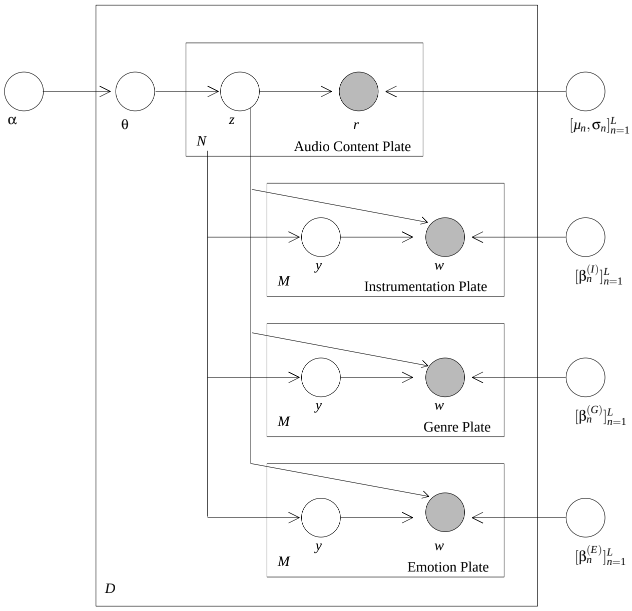
α
θ
z
r
N
Audio Content Plate
$[\mu_n, \sigma_n]_{n=1}^L$
y
w
M
Instrumentation Plate
$[\beta_n^{(I)}]_{n=1}^L$
y
w
M
Genre Plate
$[\beta_n^{(G)}]_{n=1}^L$
y
w
M
Emotion Plate
$[\beta_n^{(E)}]_{n=1}^L$
D

Figure 5: Hierarchical Corr-LDA Graphical Model for Music Annotation

5 Discussion

In this review of automatic music annotation research, we have observed that most systems pose the annotation problem as a supervised learning problem. One drawback of supervised learning is that it requires a classifier be trained for each individual musical concept (genre, instrumentation, emotion). By introducing latent variables, we can create a general model, which allows for a more flexible labeling scheme. Latent variable models have been successfully used for image annotation. We have suggested three variants of the Corr-LDA model that might be useful for music annotation.

Our second observation is that the majority of research has focused on musical feature design: extraction, selection, and integration of musical features. While this is important and will continue to be a hot research topic, it is hard to compare results since researchers do not use a standard data set. In computer vision, the Optical Character Recognition (OCR) data set of handwritten digits is one example of a standard data set that has allowed comparison of competing image feature sets. Though standard music data sets exist, such as the RWC data set[5], they are usually small or expensive. Instead, researchers use a homemade data set created from the their personal collections of music. Unfortunately, these homemade data sets cannot be shared due to copyright laws.

Another general problem is that there is no standard labeling scheme for music. In our examination of genre classification research, we note that the different assumptions about genre lead to inconsistent genre taxonomies. In all cases, the flat taxonomies are inadequate for describing the hierarchical taxonomies found used by the commercial music distributors (Amazon, ITunes, Allmusic). One benefit of using an unsupervised model is that it makes few assumptions about the musical concepts. When considering genre under the hierarchical Corr-LDA model, a song can be annotated with one of more of the keywords from the genre vocabulary.

Despite the fact that automatic segmentation of sound has been developed by computer music researchers, few studies have attempted to use song segments to improve classification. Supervised annotation approaches could use late information fusion (according the sum rule, median rule, or majority voting) to incorporate the extra information provide by each of the segments. The unsupervised approaches discussed in Section 4 do make use segments for annotation. In the Corr-LDA, the probability that a specific music keyword, such as ”guitar” or ”blues”, is used to annotated a song increases with the number of segments that exhibit similar characteristics. In addition, the model can be adapted to search-by-keyword for specific segments that have such characteristics. This will be useful if, for example, a user is interested in finding ”saxophone” solos in a large database of music.

Future research will involve applying unsupervised probabilistic models to the task of music annotation. A difficult first step will be to find a labeled data set. Once found (or created), we will use existing music segmentation algorithms to segment the music track. We will then use feature extraction and integration techniques to find feature vectors for each segment. Using the labeled data, we will train and test unsupervised model such as the Hierarchical Corr-LDA model presented in this work.

Acknowledgements

I would like to thank Charles Elkan, Shlomo Dubnov, and Dave Kauchak for their guidance throughout the research for and writing of this paper. I would also like to thank Dean Tullsen and Serge Belongie for their time and energy while on my Research Exam committee.

References

[AP03] Jean-Julien Aucouturier and Francois Pachet, *Representing musical genre: A state of the art*, Journal of New Music Research **32** (2003), no. 1, 83–93.

[BJ03] David M. Blei and Michael I. Jordan, *Modeling annotated data*, SIGIR ’03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval (New York, NY, USA), ACM Press, 2003, pp. 127–134.

[Ble04] David Blei, *Probabilistic models of text and images*, Phd. Dissertation, University of California, Berkeley (2004).

[Coo99] Perry R. Cook (ed.), *Music, cognition, and computerized sound: an introduction to psychoacoustics*, MIT Press, Cambridge, MA, USA, 1999.

[CV05] Gustavo Carneiro and Nuno Vasconcelos, *Formulating semantic image annotation as a supervised learning problem*, To appear in CVPR ’05: IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[Dau92] Ingrid Daubechies, *Ten lectures on wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[5] The RWC database can be found at http://staff.aist.go.jp/m.goto/RWC-MDB/ .

[DBT03] Roger Dannenberg, William Birmingham, and George Tzanetakis, *The musart testbed for query-by-humming evaluation*, ISMIR 03: Proceedings of the Fourth International Conference on Music Information Retrieval, 2003.

[ERD05] Slim Essid, Gael Richard, and Bertrand David, *Instrument recognition in polyphonic music*, ICASSP '05: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 2005, pp. 245–248.

[FD02] Joe Futrelle and Stephen Downie, *Interdisciplinary communities and research issues in music information retrieval*, ISMIR 02: Proceedings of the Third International Conference on Music Information Retrieval, 2002, pp. 215–221.

[Foo97] Jonathan Foote, *Content-based retrieval of music and audio*, Proceedings of SPIE: Multimedia Storage and Archiving Systems II, 1997.

[Foo99] ______, *An overview of audio information retrieval*, Multimedia Systems **7** (1999), no. 1, 2–10.

[GD04] Fabien Gouyon and Simon Dixon, *Dance music classification: A tempo-based approach*, ISMIR '04: International conference on Music Information Retrieval proceedings (Barcelona), Universitat Pomeu Fabra, Oct. 2004, pp. 501–504.

[Har98] William M. Hartmann, *Signals, sound, and sensation*, Springer-Verlag, 1998.

[HPD03] Perfecto Herrera, Geoffroy Peeters, and Shlomo Dubnov, *Automatic classification of musical instrument sounds*, Journal of New Musical Research **32-1** (2003), 3–21.

[Kau01] David Kauchak, *Audio meets image retrieval techniques*, UCSD Technical Report (2001).

[KB02] Ja-Young Kim and Nicholas Belkin, *Categories of music description and search terms and phrases used by non-music experts*, ISMIR '02: Proceedings of the Third International Conference on Music Information Retrieval (Michael Fingerhut, ed.), 2002, pp. 209–214.

[Lew02] M.S. Lewicki, *Efficient coding of natural sounds*, Nature Neuroscience **5** (2002), no. 4, 356–363.

[LLDBM04] Micheline Lesaffre, Marc Leman, Bernard De Baets, and Jean Martens, *Methodological considerations concerning manual annotation of musical audio in function of algorithm development*, ISMIR '04: International conference on Music Information Retrieval proceedings (Barcelona), Universitat Pomeu Fabra, 10 2004, pp. 64–71.

[LLZO02] Tao Li, Qi Li, Shenghuo Zhu, and Mitsunori Ogihara, *A survey on wavelet applications in data mining.*, SIGKDD Explorations **4** (2002), no. 2, 49–68.

[LO03] Tao Li and Mitsunori Ogihara, *Detecting emotion in music*, ISMIR '03: Proceedings of The Fourth International Conference on Music Information Retrieval, 2003, pp. 239–240.

[LO05] ______, *Music genre classification with taxonomy*, ICASSP '05: IEEE International Conference on Acoustics, Speech, and Signal Processing, March 2005, pp. 197–200.

[Log00] B. Logan, *Mel frequency cepstral coefficients for music modeling*, ISMIR '00: International Symposium on Music Information Retrieval (2000).

[LOL03] Tao Li, Mitsunori Ogihara, and Qi Li, *A comparative study on content-based music genre classification*, SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (New York, NY, USA), ACM Press, 2003, pp. 282–289.

[LSDM01] Dongge Li, Ishwar K. Sethi, Nevenka Dimitrova, and Tom McGee, *Classification of general audio data for content-based retrieval*, Pattern Recognition Letters **22** (2001), no. 5, 533–544.

[LT03] Tao Li and George Tzanetakis, *Factors in automatic musical genre classification of audio signals*, WASPAA '03: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2003, pp. 143–146.

[Mak75] J. Makhoul, *Linear prediction: A tutorial review*, Proceedings of IEEE **63(4)** (1975), 561–580.

[MAL05] Anders Meng, Peter Ahrendt, and Jan Larsen, *Improving music genre classification by short-time feature integration*, ICASSP '05: IEEE International Conference on Acoustics, Speech, and Signal Processing, March 2005.

[MB03] Martin F. McKinney and Jeroen Breebaart, *Features for audio and music classification*, ISMIR '03: International Conference on Music Information Retrieval, 2003, pp. 151–158.

[OSB99] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck, *Discrete-time signal processing (2nd ed.)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1999.

[Pac03] Francois Pachet, *Content management for electronic music distribution*, Communications of the ACM **46** (2003), no. 4, 71–75.

[PC00] Francois Pachet and Daniel Cazaly, *A taxonomy of musical genres*, RIAO '00: Content-Based Multimedia Information Access, 2000.

[PE04] R. Mitchell Parry and Irfan Essa, *Feature weighting for segmentation*, ISMIR '04: Proceedings of the International Conference on Music Information Retrieval, Universitat Pompeu Fabra, October 10-14 2004, pp. 116–119.

[PZ04] Francois Pachet and Aymeric Zils, *Automatic extraction of musical descriptors from acoustic signals*, ISMIR '04: International conference on Music Information Retrieval proceedings (Barcelona), Universitat Pomeu Fabra, 10 2004, pp. 353–356.

[RJ93] Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of speech recognition*, Prentice Hall, 1993.

[TC02] George Tzanetakis and Perry Cook, *Musical genre classification of audio signals*, IEEE Transaction on Speech and Audio Processing **10** (2002), no. 5, 293–302.

[TE05] Douglas Turnbull and Charles Elkan, *Fast recognition of musical genres using rbf networks*, IEEE Transactions on Knowledge and Data Engineering **17** (2005), no. 4, 580–584.

[Tza02] George Tzanetakis, *Manipulation, analysis and retrieval systems for audio signals*, 2002.

[WC04] Kris West and Stephen Cox, *Features and classifiers for the automatic classification of musical audio signals*, ISMIR '04: International conference on Music Information Retrieval proceedings (Barcelona), Universitat Pompeu Fabra, 10 2004.

[YL04] Dan Yang and WonSook Lee, *Disambiguating music emotion using software agents*, ISMIR '04: International conference on Music Information Retrieval proceedings (Barcelona), Universitat Pomeu Fabra, 10 2004, pp. 218–223.

[ZE02] Bianca Zadrozny and Charles Elkan, *Transforming classifier scores into accurate multiclass probability estimates*, SIGKDD '02: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2002, pp. 694–699.

A APPENDIX A: Digital Signal Processing Background

The material covered in this appendix is intended to give the reader some background on digital signal processing concepts that were introduced in this paper. The reader may wish to refer to [OSB99] and [RJ93] for a more detailed review of these concepts.

A.1 Digital Audio Signals

When music is recorded, the pressure from the acoustic wave is measured using a microphone. These measurements are taken at regular time interval and each measurement is quantized. (See Figure **??** The result is a bitstream that can be analyzed using digital signal processing techniques. For example, CD audio is sampled 44,100 times per second where each sample is represented as a 2 8-bit (2-channel stereo) values in the range $[2^{-7}, 2^{7}]$. A sound represented as a time series of pressure measurements is representation in the *time domain*.

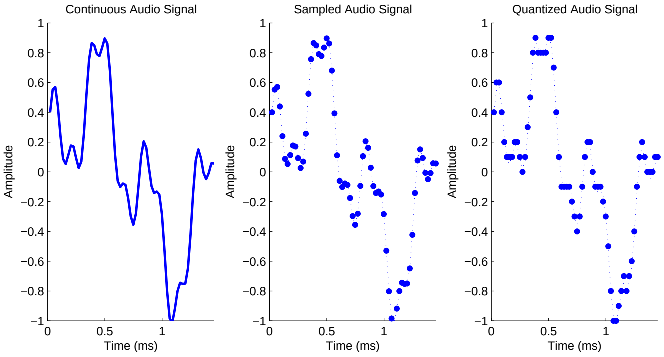

Figure 6: Music is a continuous signal (left) that is sampled (center) and quantized (right) so that it can be represented as a bitstream.

Sound can also be represented as the sum of sinusoids (i.e. a set of sine and cosine waves oscillating at different frequencies.) That is, a signal of N samples can be written as

$$x = \sum_{k=0}^{N/2} a_k^{(r)} \cos(2\pi(\frac{k}{N})) + a_k^{(i)} \sin(2\pi(\frac{k}{N})). \tag{2}$$

We can represent the signal in the *frequency domain* using the coefficients $\{(a_1^{(r)}, a_1^{(i)}), ..., (a_{N/2}^{(r)}, a_{N/2}^{(i)})\}$. The *magnitude* and *phase* of the k-th frequency component are given by

$$X_M[k] = \sqrt{(a_k^{(r)})^2 + (a_k^{(i)})^2} \tag{3}$$

$$X_P[k] = \arctan(\frac{a_k^{(i)}}{a_k^{(r)}}) \tag{4}$$

In general, magnitude information is used during feature extraction while the phase information is ignored. The argument from ignoring phase information is based on the results of perceptual studies of human hearing that show phase information to be relatively unimportant when compared to magnitude information [Log00].

For example, consider the waveform $\mathbf{x}$ shown on the left panel of Figure 7. Each sample of $\mathbf{x}$ is found according to

$$x[i] = 0.55\sin(2\pi f_1 i) + 0.4\cos(2\pi f_2 i) + 0.15\sin(2\pi f_3 i)$$

where f_1, f_2, and f_3 are 687 Hz, 2.28 kHz, and 6.46 kHz, respectively. The frequency domain representation (right panel), called the *spectrum* of the signal, shows three distinct peaks, one for each of the three non-zero frequency components. Note that the relative heights of these peaks correspond to the amplitudes (i.e. 0.55, 0.40, 0.15) of the three sinusoids. (We will discuss how the heights of the peaks (i.e. the coefficients $\{(a_k^{(r)}, a_1^{(i)})\}$) are found in the next subsection when we discuss the discrete Fourier transform.)

A number of spectral-shape feature (TC02, MB03) are extracted from the spectrum. For example, *Spectral Centroid* is the center of gravity of the spectrum and is given by

$$C = \frac{\sum_{k=1}^{N/2} X_M[k] * k}{\sum_{k=1}^{N/2} X_M[k]}.$$

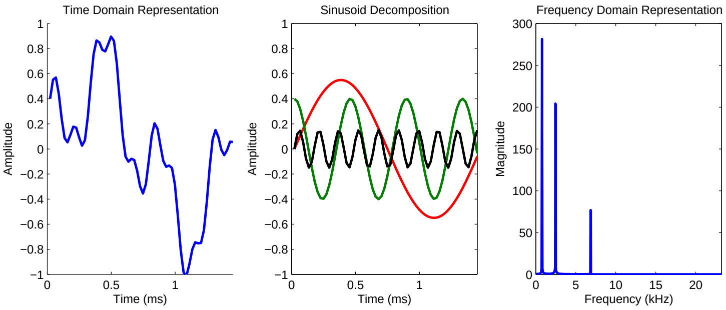

Figure 7: Time and frequency domain representation of an audio signal sampled at 44,100 samples per second. The audio signal in represented in the time domain (left) can be broken into a sum of sinusoids (middle), and then represented in the frequency domain (right) using the magnitudes of the frequency components.

The Spectral Centroid can be though of as a measure of 'brightness' since songs are consider brighter when they have more high frequency components. A full description of other features, such as RMS Energy, Spectral Rolloff, and Spectral Flux, are described in [Tza02].

A.2 Time-Frequency Domain Transforms

In the previous section, we introduced the concept of time and frequency domain representations of an audio signal. In this section, we will describe in detail both the real discrete Fourier transform (DFT) and the real short-time Fourier transfrom (STFT), and introduce the discrete cosine transfrom (DCT), the discrete wavelet transform (DWT), and the gammatone transform (GT). (We will look at *real* transforms, rather than complex transforms, since music is always a real-valued time series and we are only concerned with positive frequencies. For more information on complex transforms, see [OSB99].) All of these transforms mapped a time series of samples into a frequency domain or time-frequency domain representation. They differ in that they use different sets of *basis functions* and *projection operators* to perform the transformation.

Given a signal **x** with N samples, the basis functions for the DFT will be N/2 sine waves and N/2 cosine waves that correspond to the coefficients $\{(a_k^{(r)}, a_k^{(i)})\}$ introduced in the previous subsection. The projection operator is correlation, which is a measure of how similar two time series are to one another. The coefficients are found by

$$a_k^{(r)} = \frac{2}{N} \sum_{i=0}^{N-1} x[i] \cos(2\pi \frac{k}{N} i) \tag{5}$$

$$a_k^{(i)} = -\frac{2}{N} \sum_{i=0}^{N-1} x[i] \sin(2\pi \frac{k}{N} i) \tag{6}$$

$$. \tag{7}$$

In practice, the DFT is implemented by the fast Fourier transform (FFT), which can computes the exact coefficients in computationally efficient manner. The inverse DFT (iDFT) transform, which maps the spectrum to a time series, is given by equation 2. Another commonly used transform, the discrete cosine transform (DCT), is a transform like the DFT, but uses a basis of cosine function that differ in frequency by multiples of π. Note that the DFT uses a basis of sine and cosine function that differ in frequency by multiples of 2π.

One drawback of both the time series representation and the spectrum representation is that neither simultaneously represents both time and frequency information. A time-frequency representation is found using the short-time Fourier transform (STFT). First, the audio signal is broken up into a series of (overlapping) segments. Each segment is multiplied by a *window function*, which is usually a bell shaped curve that is non-zero over the segment and zero otherwise. The length of the window is called the *window size* and

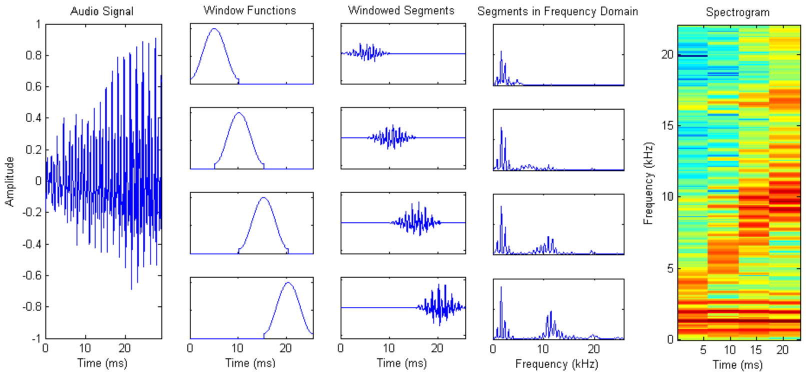

Figure 8: STFT Example: An audio signal (far left) in the time domain is broken into windowed segments (middle). Each segment is transformed to the frequency domain using the DFT (right). The final representation, called a spectrogram, contain both time and frequency information.

the amount of overlap is called the *hop size*. The real DFT is defined as

$$a_{n,k}^{(r)} = \frac{2}{N}\sum_{i=0}^{N-1} x[n+i]w[i]\cos(2\pi\frac{k}{N}i) \quad (8)$$

$$a_{n,k}^{(i)} = -\frac{2}{N}\sum_{i=0}^{N-1} x[n+i]w[i]\sin(2\pi\frac{k}{N}i) \quad (9)$$

$$. \quad (10)$$

where $w[i]$ is the window function. In this representation, the coefficients $\{(a_{n,k}^{(r)}, a_{n,k}^{(i)})\}$ are indexed by both time and frequency.

There are two important ideas involved with the STFT: the more samples in a windowed segment, the better the frequency resolution, and the shorter the window length, the better the time resolution. This creates a trade-off between time and frequency resolution. The STFT is know as a *fixed resolution* transform since the frequency and time resolution is the same for each frequency band. (A frequency band is a range of frequencies.)

An alternative to the STFT are transforms from the family of wavelet transforms, all of which have the property of *variable resolution*. That is, the signal is broken up into different frequency bands, and each band is analyzed using basis functions that match the scale of the band. High frequency bands are correlated with with shorter waves giving good time resolution. Low frequency bands are correlated with longer waves giving better frequency resolution. This idea is illustrated in Figure 9. While a complete mathematical description of the wavelet transform is beyond the scope of this Appendix, the reader may wish to refer to [LLZO02] which provides an introduction to wavelet analysis for data mining applications.

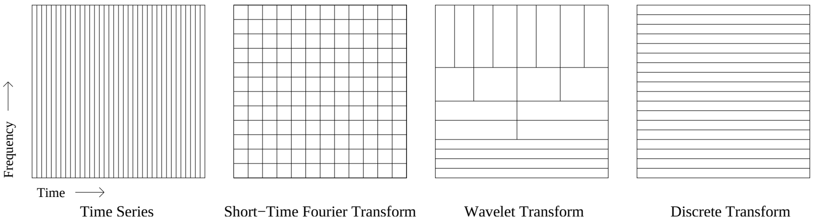

Figure 9: Trade-off between time and frequency resolution: The best time resolution is found in the time series representation, while the best frequency resolution is found using one Fourier transform on the entire signal. Using STFT and Wavelet transforms result is different time-frequency representations.

Lastly, the family of Gammatone transforms provides variable resolution time-frequency representations of an acoustic signal that are designed to model the human auditory system. More information on

Gammatone filters can be found in Chapter 10 of [Har98].

A.3 Mel-frequency Cepstral Coefficients (MFCC)

The most common set of features used in speech recognition and music annotation systems are the Mel-frequency cepstral coefficients (MFCC). MFCC are short-time features that characterize the magnitude spectrum of an audio signal. For each short-time (25 ms) segment, the feature vector is found using the five step algorithm given in Algorithm 1. The first step is to obtain the magnitude of each frequency component in the frequency domain using the DCT. We then take the log of the magnitude since perceptual loudness has been shown to be approximately logarithmic. The frequency components are then merged into 40 bins that have been space according the Mel-scale. The Mel-scale is mapping between true frequency and a model of perceived frequency that is approximately logarithmic. Since a time-series of these 40-dimensional Mel-frequency vectors will have highly redundant, we could reduce dimension using PCA. Instead, the speech community has adopted the discrete cosine transform (DCT), which approximates PCA but does not require training data, to reduce the dimensionality to a vector of 13 MFCCs.

A more complete treatment of MFCC feature extraction, including theoretical and empirical justification for music modeling, can be found in [Log00].

Algorithm 1 Calculating MFCC Feature Vector

```
1: Calculate the spectrum using the DFT
2: Take the log of the spectrum
3: Apply Mel-scaling and smoothing
4: Decorrelate using the DCT.
```

A.4 The Autoregressive (AR) Model

An autoregressive model of order P assumes that a sample $x[n]$ at time n can be predicted using past samples according to the model

$$x[n] = (\sum_{i=1}^{P} a_i x[n-i]) + \varepsilon[n]$$

where the a_i's are called the linear predictive coding (LPC) coefficients and $\varepsilon[n]$ is a random variable with mean 0 and variance σ^2. The LPC coefficients are found by minimizing variance of $\varepsilon[n]$ on a training time series and can be calculated using the Levinson-Durbin algorithm [Mak75]. The intuition behind the autoregressive model is that the time series have been generated by passing white noise through an all-pole filter that 'colors' the signal. The LPC coefficients $\{a_1, ..., a_P\}$ are informative features since they characterize the spectrum of the coloring filter.

Future Thesis Committee Members

Potential future committee members is included Charles Elkan, Shlomo Dubnov, Sanjoy Dasgupta, Gary Cottrell, Serge Belongie, Nuno Vasconcelos, Alon Orlitsky, and Bhaskar Rao.

Future Work

- Feature Design
  - Short-Time Extraction
    - Wavelets
      - Read Li's Wavelet Tutorial - wavelet.org
      - Compare with STFT with multiple time resolutions (Serra CCRMA) - similar to Ullman's Intermediate Complexity Feature for Vision
    - Gammatone Filters (MB03) - read Hartmann *Signals, sound, and sensation*).
    - Marsyas Features (TC02) - use filterbank transform(MB03) and Autoregression (MAL05) on Marsyas features
- Modeling
  - Using Music Segments
    - Compare block-based and automatic segmentation
    - Read automatic segmentation literature - self-similarity, changes in short-time features
    - Vision - Apply Nuno's Weibull and Erlang priors to model time since last segment.
  - Latent Models
    - Implement Blei's Unsupervised Corr-LDA - parameter estimation with variational EM
    - Implement Nuno's Supervised M-ary - parameter estimation with Mixture Hierarchy.
    - Review other image annotation literature on latent models
- Data Sets
  - Amazon Free Downloads - Artist, Album, Song Title, Genre (3 level hierarchy).
  - RWC Database - small and expensive
  - Homemade Database - hand labels for instrumentation, genre, emotion, etc.
  - Cross reference AMG Allmusic, Moodlogic, etc to text mine for emotion, instrumentation, similar songs, etc.