#jazz : Automatic Music Genre Detection

Tom Camenzind
tcamenzi@stanford.edu
Department of Computer Science
Stanford University, CA

Shubham Goel
gshubham@stanford.edu
Department of Computer Science
Stanford University, CA

Abstract

Automatic genre classification of music is an important topic in Music Information Retrieval with many interesting applications. A solution to genre classification would allow for machine tagging of songs, which could serve as metadata for building song recommenders. In this paper, we investigate the following question:

Given a song, can we automatically detect its genre?

We look at three characteristics of a song to determine its genre: timbre, chord transitions, and lyrics. For each method, we develop multiple data models and apply supervised machine learning algorithms including k-means, k-NN, multi-class SVM and Naive Bayes. We are able to accurately classify $65 - 75\%$ of the songs from each genre in a 5-genre classification problem between Rock, Jazz, Pop, Hip-Hop, and Metal music.

1. Introduction

1.1. Motivation

There has been an explosion of musical content available on the internet. Some sites, such as Spotify and Pandora, carefully curate and manually tag the songs on their sites [1]. Other sources, such as Youtube, have a wider variety of music, but many songs lack the metadata needed to be searched and accessed by users. One of the most important features of a song is its genre. Automatic genre classification would make hundreds of thousands of songs by local artists available to users, and improve the quality of existing music recommenders on the web.

1.2. Past Work

Detecting a song's genre from its raw waveform is difficult, and has been studied in a number of previous papers. Many characteristics of music that humans recognize in music – beat, chord progressions, and distinct instruments – cannot be reliably detected from audio files [2]. Without these intermediate-level features, we cannot approach the genre classification problem in the same way a trained musician would. Previous work has focused on applying existing signal processing techniques to find low-level features that correlate with musical genre. In particular, Mel-Frequency Cepstrum (MFC) coefficients – originally used in voice recognition tasks [3] – have proved particularly useful in describing a song's timbre or tone quality [4].

1.3. Our Work

We use three techniques to determine a song's genre.

- We analyze a song's timbre using MFC coefficients. MFC coefficients represent the power spectrum of a short-duration sound wave, and are scaled to more closely match a human's perception of sound [5]. Each of the 12 coefficients corresponds to some quality of the sound– its loudness, tone brightness, sharpness of the soundwave, and so forth [6]. We investigate three ways of modeling a song as a series of MFC coefficients. First, we use the method described in [7][8] and model our data using a Multivariate Gaussian Distribution. We also explore techniques we call Timbre Vector Voting and Time Window Gaussians.
- We analyze the chord transitions within a song. Because reliably detecting chords within a song is still an open research question, we focus on modeling only the root of the chord. We formulate the chord transitions as a Markov process, and calculate the MLE Markov process used to generate our song.
- We analyze a song's lyrics. Although lyrics do not define the music itself, in practice there is a strong correlation between a song's lyrics and its genre [9].

2. Dataset

We used a subset of 10000 songs from the Million Songs Dataset [10], a freely available collection of audio features and metadata for a million contemporary popular music tracks. The dataset provided features describing the song's timbre at 250 millisecond intervals. Specifically, each interval had 12 MFC coefficients calculated. The dataset also lists

the dominant chord being played at every 250 millisecond interval. Each song also has associated tags, which we analyzed to determine the genre.
The Million Songs Database did not contain complete song lyrics, and so we gathered our own data. We wrote a crawler to download song lyrics from `songlyrics.com` [11], which has a "Top 100" category for country, hip hop/rap, R&B, rock and pop. We parsed the lyrics of 500 songs to construct our vocabulary.

3. Methodology

3.1. Timbral Analysis

Here, we analyze a song's timbre to determine its genre. At every 250 millisecond interval in the song, we have 12 MFC coefficients, calculated as follows.
We begin by taking the Fourier Transform of the song waveform :

$$X_{2\pi}(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-i\omega n}$$

We scale the result using the Mel scale, which models the sensitivity of the human ear to sound frequencies. We take the discrete cosine transform of the result, which is defined as follows. For $k = 0, \ldots, N-1$,

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos[\frac{\pi}{N-1}nk]$$

There are 1000 intervals in a typical song, for a total of $12 * 1000 = 12000$ features describing a song. This is too many features for our training set; in practice, we can hold only 1000 songs in RAM at once, so we analyze 200 songs per genre. In all analyses of timbre and chords, we train on 70% of songs and test on 30%. We experimented with several models to reduce the number of features per song.

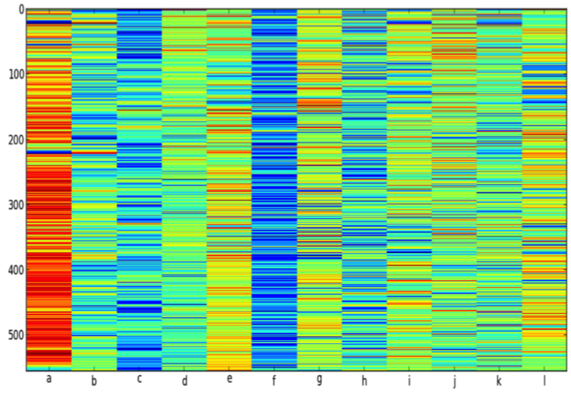

Figure 1. MFCC Representation of a song. The x-axis shows each of the 12 MFC coefficients; the y-axis shows each time interval.

3.1.1. Multivariate Gaussian Model

Here, we assume that the timbral coefficients of the song at each time interval are drawn from a Multivariate Gaussian distribution. We calculate the Maximum Likelihood Estimation Gaussian for each song, and represent the song using the mean μ and covariance Σ of this distribution [7][8]. This reduces the number of features from $12,000$ to $12 + 12 * 12 = 156$ features.

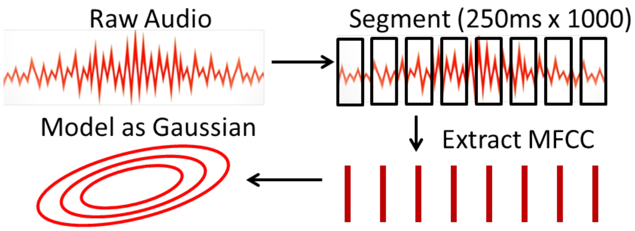

Figure 2. MFC Coefficient Extraction and the Gaussian Model.

Multivariate Gaussian Distance Metric: Symmetrized KL Divergence
Consider two multivariate Gaussian distributions $p(x)$ and $q(x)$ with mean μ_p, μ_q and covariance matrices Σ_p, Σ_q respectively. The KL divergence [7][8] is then given by

$$2K(p||q) = \log\frac{|\Sigma_q|}{|\Sigma_p|} + tr(\Sigma_q^{-1}\Sigma_p) + (\mu_p - \mu_q)^T \Sigma_q^{-1}(\mu_p - \mu_q) - x$$

where x denotes the dimension of the feature vector. Since the KL divergence is asymmetric with respect to the distributions [7][8], our distance metric is given as

$$D(p||q) = K(p||q) + K(q||p)$$

Multivariate Gaussian Classification: k-Nearest Neighbors
As a simple test of our model's effectiveness, we implement k-nearest neighbors. To classify a new song, we calculate its MLE Gaussian, and compare that to the songs in our training set using the KL Divergence distance metric.

Multivariate Gaussian Classification: k-Means with KL divergence
We expect that the Gaussians for a single genre will be clustered together, or that there may be multiple clusters representing subgenres. For jazz, we may have one cluster associated with swing songs, others with jazz ballads, etc. To identify these clusters, we group all songs from a genre, calculate the MLE Gaussians, and run k-means using the KL divergence as our distance metric. We do this for all genres, and store all the centroids calculated. To classify a new song, we find the nearest centroid, and assign the song to the genre of that centroid.
Interestingly, we obtained the highest classification accuracy using $k = 1$, meaning each genre was best modeled by a single cluster. This suggests that songs of the same genre have tightly related timbres, as opposed to being split into several subgenres with distinct timbres.

3.1.2. Timbre Vector Voting

By modeling the entire song as a single Gaussian, we lose information about the individual feature vectors. We expect that some timbral vectors are highly indicative of genre (for example, those extracted during an electric guitar solo will be characteristic of Metal) while others will be common to many genres. We would like to automatically distinguish between these types of feature vectors during classification.

Instead of training a classifier to distinguish between songs, we train our classifier to detect which genre a timbre vector most likely comes from. Our classifier will also produce a confidence value, indicating how strongly that timbre is associated with the predicted genre. Timbres common to many genres will have confidence values near zero.

Once we can predict where individual timbre vectors come from, we can make predictions for entire songs. For a song we want to classify, we calculate how strongly its timbre vectors are associated with a genre, and make predictions using the following.

Consider the set of genres G and the set of timbre vectors (V_s) for a song s. Our classifier has a `confidence` function, that returns a value indicating how strongly a timbre vector is associated with the given genre. The predicted genre (P_s) for that song is given by:

$$P_s = \arg\max_{g \in G} \sum_{v \in V_s} \texttt{confidence(v;g)}$$

Each timbre vector "votes" for what genre vector it is associated with. More weight is given to high-confidence associations, and the genre with the most "votes" across the songs timbre vectors wins.

We now describe which classifiers we used to predict genre associations.

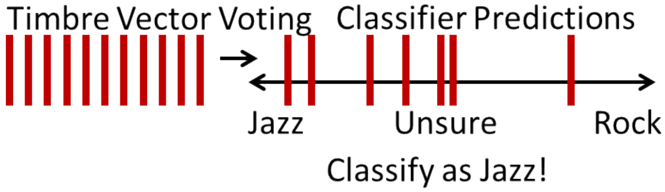

Figure 3. Timbre Vector Voting. We predict the genre of each timbre vector, and take the confidence-weighted average to predict the genre of a song.

Timbre Vector SVM

Here, we train a multi-class SVM on the timbre vectors, each labeled with its genre. We can make genre predictions on timbre vectors as usual. For binary classification problems, we use the distance from the decision boundary to estimate confidence values. For multiclass problems, the SVM we used only provided 0/1 predictions, so we set the confidence to be 1 for the predicted genre, and 0 for all other genres.

k-Means Classification

For each genre, we gather all timbre vectors occurring in songs associated with that genre, and run k-means on the timbre vectors. We store the centroids calculated for all the genres. To classify a new timbre vector, we find the nearest centroid in each genre, and calculate the distance to that centroid. The negative of this distance is the confidence measure for each genre, so the song will be assigned to the genre whose centroids best fit the song's timbre vectors.

3.1.3. Time Window Gaussians

With timbre vector voting, every timbre vector votes independently. In practice, we expect that neighboring timbre vectors will be closely related. Here, we propose a technique that captures the dependencies between timbre vectors (the benefits of our Multivariate Gaussian model), while providing the finer time resolution of timbre vector voting.

Here, we group neighboring timbre vectors (groups of 10) to form time windows. We calculate the MLE Multivariate Gaussian distribution for each time window. To avoid overfitting, we restrict the covariance matrix to be diagonal. Now, we can represent each time window as a vector containing the mean and (diagonal) covariance matrix, which we call the Time Window Gaussian.

We now use the same methods of Timbre Vector Voting, but replacing the timbre vectors with the Time Window Gaussians. The mean vector provides the fine time resolution we had with Timbre Vector Voting, while the covariance matrix also captures the structure of the timbre changes over the time interval.

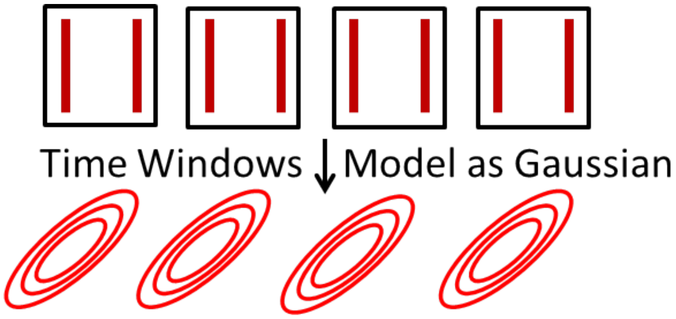

Figure 4. The Time Windows Gaussian Model. We group neighboring timbre vectors, and calculate the MLE Gaussian for each group.

3.2. Chord Transitions Markov Model

For trained musicians, chord progressions are one of the defining features of musical genre. We investigate whether they can help us with our automatic classification task. Previous research has successfully detected chord progressions from individual instruments [12]. However, detecting chords from ensemble music is a difficult, open research problem [13]. In particular, background noise – especially from percussion – makes it impossible to reliably detect chords using current signal processing techniques. Therefore, we restrict our efforts to looking at the dominant note (the chord's root) at each time interval, and finding transition probabilities between time intervals. This can provide a variety of information about a song. This can correlate with tempo (slow songs will have 0-chroma transitions more often), major vs. minor (3-chroma vs 4-chroma transitions), and so forth. We look at transitions between 250 ms time intervals.

We assume that each song is generated according to a Markov process, and that different genres will have different Markov transition probabilities. All transitions are in the range $[0, 11]$ corresponding to the 12 chromatics in an octave.

Define procedure

$$\texttt{getTransition}(\texttt{s}, \texttt{t1}, \texttt{t2}) = \\ (\texttt{dominantChroma}(\texttt{s}, \texttt{t1}) - \texttt{dominantChroma}(\texttt{s}, \texttt{t2})) \bmod 12$$

where `s` is a song, and `t1` and `t2` are time intervals. The function `dominantChroma` returns a number $0, \ldots, 11$ corresponding to C, C#, . . ., B and mod 12 maps this change into our octave of interest.

Consider the number of transitions for a song s, N_s. The ith transition probability, T_i, is then given by:

$$T_i = \frac{1}{N_s} \sum_{t=1}^{N_s} 1\{\texttt{getTransition}(\texttt{s}, \texttt{t}, \texttt{t} + 1) = i\}$$

We calculate this for every possible interval to get a chord transitions vector representing the song.

Once we have modeled the songs, we train an SVM on our Chord Transitions vectors with genre labeled, and classify new songs by calculating their Chord Transitions vector and making predictions using this SVM.

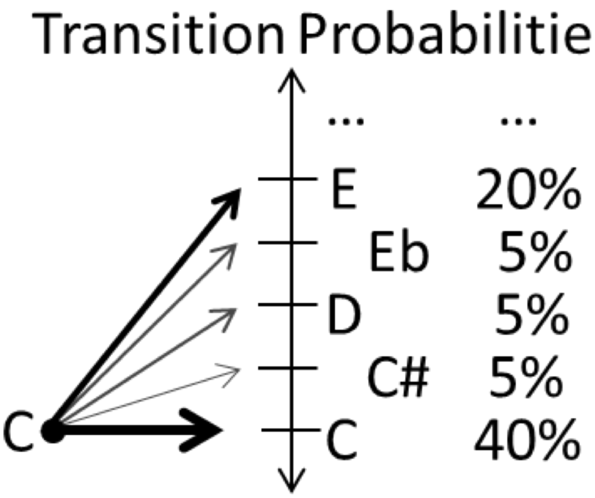

Figure 5. The transition probabilities calculated for a song.

3.3. Lyrics

3.3.1. Model

All song lyrics are parsed to produce a vocabulary. Each song's lyrics are converted into a multinomial model representation that consists of the indices of words in the vocabulary and frequencies of each index.

3.3.2. Multinomial Naive Bayes

We implemented multinomial Naive Bayes. We calculate the probability of a training example occurring given each genre. The parameters for each class are given as [9], $l = 1, 2, \ldots, 5$

$$\phi_{k|y=l} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = l\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = l\} n_i + |V|}$$

$$\phi_{y=l} = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = l\}}{m}$$

Once the probabilities and parameters are trained, the probability for test data is calculated and decisions are made based on max-likelihood estimates of the training data belonging to the genre. For lyrics, we train on 85% of songs and test on 15%.

4. Results

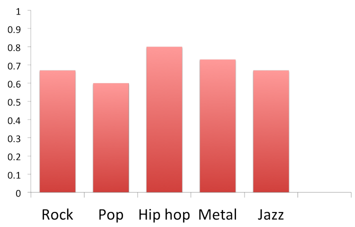

Figure 6. (a)

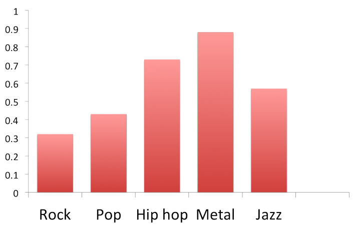

Figure 7. (b)

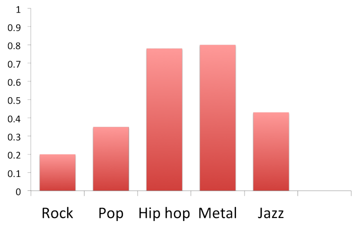

Figure 8. (c)

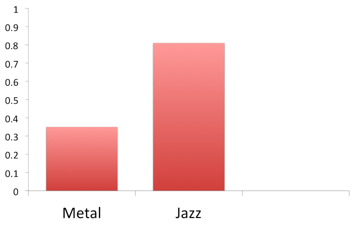

Figure 9. (d)

(a) k-Means Single Gaussian, $k = 1$ (b) k-Means Gaussian Window, $k = 4$
(c) SVM Gaussian Window (d) Chord Markov Model SVM

Each bar in the chart gives the percent of songs of that genre that were classified as being that genre. That is, these graphs represent the diagonal of each confusion matrix.

Timbral analysis using MFC coefficients provided the best method for predicting musical genre, where we achieved a 70% accuracy on a 5-way classification problem. This supports the results of previous work on this subject. Classification by lyrics also performed well. Examining chord transitions worked well for binary classification problems, but for the 5-way classification problem posed here, there was too little separation among the data to obtain accurate predictions.

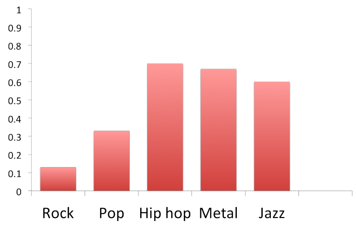

Figure 10. (e)

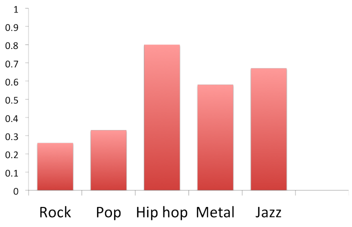

Figure 11. (f)

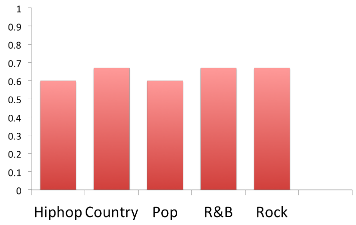

Figure 12. (g)

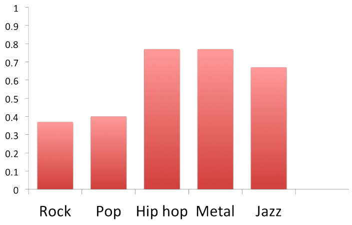

Figure 13. (h)

(e) Single Feature Vectors SVM (f) k-NN Single Gaussian, $k = 10$
(g) Lyrics (h) k-Means Single Feature, $k = 8$

5. Challenges and Future Work

The field of Music Information Retrieval is growing and has huge potential. While text-based information is readily queryable, music is a major source of online content that has not been unlocked. Currently, genre detection is difficult because many of the high-level ideas that humans use to determine genre, such as chord progressions, cannot be determined reliably using existing signal processing techniques. Advances in signal processing techniques – in particular, ones for reliable chord, beat, and instrument detection – will allow us to better analyze and organize musical content.

Our work presents several successful methods for genre classification. By analyzing a song's timbre, chord roots, and lyrics, we obtain accuracies of up to 70% on 5-way classification problems. This could prove commercially valuable for song databases too large to manually tag all songs. Our predictions could be used as a feature in song database relevance rankings, to give higher weight to songs that are likely a genre of interest.

We have several ideas to improve our classifier in future work. Other features used in voice recognition – for example, the zero-crossing rate and the short-term sound power spectrum – may also correlate with musical genre and enhance our classification accuracy.

6. References

1. Panagakis Y. and Kotropoulos C. "Music Classification by Low-Rank Semantic Mappings". *EURASIP Journal on Audio, Speech and Music Processing* (2013)

2. Weller A., Ellis D. and Jebara T. "Structured Prediction Models for Chord Transcription of Music Audio". *Columbia University* (2010)

3. Milner B. and Shao X. "Speech Reconstruction from Mel-Frequency Cepstral Coefficients Using a Source-Filter Model". *University of East Anglia, UK* (2011)

4. Silla C., Koerich A. and Kaestner C. "A Machine Learning Approach to Automatic Music Genre Classification". *Journal of the Brazilian Computer Society* (2008)

5. Tzanetakis G. and Cook P. "Musical Genre Classification of Audio Signals". *IEEE Transactions on Speech and Audio Processing* (2002)

6. Jahan, T and DesRoches, D. "The Echonest Analyzer Documentation" `http://docs.echonest.com.s3-website-us-east-1.amazonaws.com/_static/AnalyzeDocumentation.pdf` (2009)

7. Rajani, M. and Ekkizogloy L. "Supervised Learning in Genre Classification". *Department of Computer Science, Stanford University* (2009)

8. Haggblade, M., Hong Y. and Kao K. "Music Genre Classification". *Department of Computer Science, Stanford University* (2011)

9. Rabee A., Go K. and Mohan K. "Classifying the Subjective: Determining Genre of Music from Lyrics". *Department of Computer Science, Stanford University* (2012)

10. Mahieux T., Ellis D., Whitman B., and Lamere P. "The Million Song Dataset". *In Proceedings of the 12th International Society for Music Information Retrieval Conference* (2011)

11. Lyrics Dataset for Songs. `www.songlyrics.com`

12. Arndt C. and Li L. "Automated Transcription of Guitar Music". *Stanford University* (2012)

13. Jiang N., Grosche P., Konz V. and Muller M. "Analyzing Chroma Feature Types for Automated Chord Recognition" *AES International Conference, Germany* (2011)