Waveform-Based Musical Genre Classification

Charles Tripp Hochak Hung Manos Pontikakis

Abstract

For a human, recognizing the genre of a piece of music is usually an effortless and thoughtless task; for a computer, genre classification is not a simple task. Previous research on this topic has found it to be a difficult machine learning problem. We have carefully chosen relevant features and an appropriate classification algorithm which achieve high accuracy genre classification. Features are extracted via spectral and time domain analysis, and then the LogitBoost algorithm is used to build an effective classifier for the data. This paper discusses the final feature set, why we chose those features, our final classification algorithm, and why we chose it.

Introduction

In this project we have implemented an automatic musical genre classification system which extracts its features directly from the music waveform. Understanding which statistics and aspects of a musical piece are most relevant to classifying its genre is an important problem for both musical database organization and music recommendation systems. We have compiled a dataset of over 2,600 songs, each belonging to a specific musical genre, and extracted a number of carefully chosen features from each song. In order to achieve a broadly applicable classification system, we have chosen to classify music between five distinct and large musical genres: rock, hip-hop, techno, classical, and pop. After a careful exploration of various features, and classification algorithms, we have arrived at a set of 102 relevant features which allow the LogitBoost algorithm to achieve an overall classification accuracy of 83%. Our features were chosen based on recommendations of related work [1][2][3][4], as well as careful hands-on analysis of the spectral and time-domain properties of songs from each genre. If an effective classification system can be created, it can be used to automatically classify songs by genre, or with slight modifications, recommend songs similar in style to a given sample of songs. Potential applications of such a system include improved automatic musical programming for radio stations, improved musical recommendation systems for vendors, and musical databases which allow searching based on various musical qualities.

Related Work

Previous work on this problem is limited, and many studies have either classified based on notational (e.g. MIDI) data [5][6], or were limited to very specific recognition tasks (identifying classical instruments from a piece of classical music) [7]. However, many previous attempts which used the waveform data have focused almost exclusively on one type of feature and analysis. For example, [2] and [8] use only the spectral characteristics of windows of the song, while [3] uses only gross spectral and time-domain statistics of the entire song. As one might suspect, a plethora of features have been proposed, but few have proven to be consistently helpful in genre classification.

Critical Features

Upon examination, a classification problem such as this requires a careful choice of the relevant features. While a four-minute long, 16-bit recording of a song requires over 20MB of storage space, a practical classification algorithm, without a very large training set, can only handle perhaps several tens of features per song. Therefore, in order to create a practical algorithm, a small number of statistics which are critical to identifying the genre of a song must be found. Features which have been particularly useful in previous work were explored, as well as several new features. In the end, a selection of both new and old features proved to be the best for our problem.

Many previous papers have suggested that vocals can be fairly easily detected from a song by measuring a statistic known as the **Zero-Crossing Rate (ZCR)** [3] [4] [8]. The ZCR of a waveform is merely the average number of times that the waveform passes through zero per second. Vocal tracks tend to have high ZCR's due to the dominance of the singer's voice in the waveform (and the dominance of frequencies in the low kilohertz range in the singer's voice). Singing (as well as most speech) tends to have waveforms with high ZCR's, so it is natural for vocal music to also have high ZCR's. In general, the ZCR is an indicator of the overall dominant frequency throughout the length of a signal. Our experimentation has shown ZCR to be one of the most important features in genre classification.

Additionally, most successful designs have used **gross spectral measurements** of a piece of music [3]. Knowing the shape of the frequency-power distribution is a critical component to identifying the genre of a song, as musical genres have distinct overall spectral shapes. This can be seen in Figure 1, the power spectrum of Chopin's Nocturn no. 2, and 50 Cent's Candy Shop. The two songs of different genre's have strikingly different spectrums. These differences are representative of the genre's as a whole. Rap music has strong beats and loud bass, resulting in a number of large low-frequency peaks. On the other hand, this instrumental classical piece has a series of harmonic peaks at midrange frequencies which are characteristic of string and wind instruments.

Our experimentation has found that a useful combination of these are: mean, median, maximum power, and the variance of the power distribution across the spectrum. These features give a simplified representation of the spectral power distribution of the song, and have proven themselves to be quite useful in separating songs by genre. For instance, classical music tends to have a low maximum power due to its broad varieties of instruments, while techno tends to have a very high maximum power due to its restricted selection of instruments and highly repetitive structure.

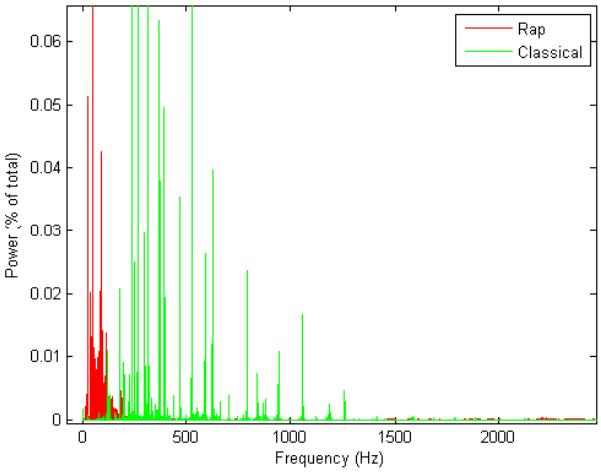

Figure 1: The power spectrum of two songs.

In addition to these simple measurements of power distribution, a slightly different set of features was selected to further hint at the overall texture of the frequency distribution of a song. We decided to logarithmically partition the frequency spectrum into 24 bands from 20Hz to 16kHz, and report the relative energy within each band. A logarithmic partitioning scheme is natural due to the fact that the human ear has a logarithmic sensitivity to frequency, as well as the fact that the musical scale scales logarithmically in frequency. Frequencies below 20Hz were not reported because the human ear cannot detect them, and frequencies above 16kHz were not reported because the sample music was only encoded up to this frequency. These features are the most significant features we encountered.

Intuitively, the **beats per minute (BPM)** of a song is an important statistic for identifying the genre of a piece of music; this intuition has been confirmed by previous work [3] [9] [10]. Many genres tend to stay within a certain range of tempos, and the BPM of a song indicates that tempo. There are several known methods, such as comb filters, for detecting the overall BPM of a piece of music, but as suggested in [12], one of the most useful BPM statistics are extracted on a per-band basis. Our implementation subdivides the song into over-

lapping 1/8th second Hanning windows (achieving a resolution of 16 windows/second). The power spectrum of each window is calculated and then further subdivided into the same frequency bands as used for the spectral shape features. Each band then uses a common moving-average plus threshold system to detect the mean and variance of the BPM within each band. This system allows the observation of not only the tempos and speed of the music within each frequency range, but also how steady that tempo is. These statistics are very helpful in separating genres. For instance, classical music tends to have a wandering melody, which shows up clearly as a high BPM variance. On the other hand, techno music has steady beats which show up as very low BPM variances. Hip-hop tends to have very high BPMs in comparison to the other genres, especially at lower frequencies.

Finally, in order to grasp the overall qualities of the dominant musical instruments of a song, we have implemented a set of features which have not been previously investigated. Some previous work in the area suggested that the **linear coefficients** resulting from fitting a simple linear model to the song might be helpful [2], but it does not appear that this has been attempted. The shape of this linear model indicates several properties of the dominant instruments used in the song which are normally difficult to extract. These properties include: pitch, attach, decay, reverberation, and timbre. For our system, we used stochastic gradient descent to fit a linear model using the previous 2048 samples of the song (the songs used a 44.1kHz sampling rate) to predict the next sample. In order to reduce the number of features sent to the classification algorithm, the power spectrum of the resulting coefficients was subdivided into the same 24 frequency bands, and the energy within each band was reported. These statistics clearly lost some detail, including phase information; however, they still proved to be quite useful and allowed our system to achieve an 8% higher classification accuracy than without using them.

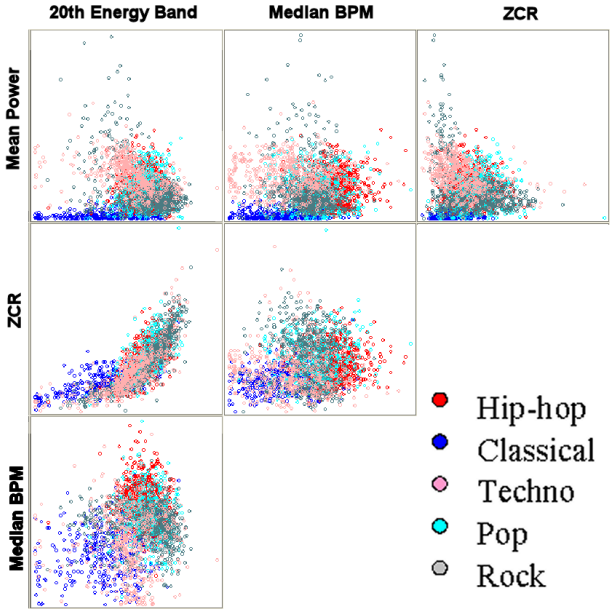

Figure 2: Scatterplots of several distinct features

Genre Classification

Having compiled this rich dataset of musical features for each song, we have applied multi-class classification techniques to recognize these 5 distinct categories of songs. In all of our experiments we have used 10-fold cross-validation to train and test.

Firstly, we tried the classic k-means algorithm to create 5 centroids of genres, and then we classified the songs in the test set by computing the Euclidean distance between the song and each centroid and then classifying the songs to their nearest centroid. This approach did not yield to good results, as we only achieved a classification accuracy of 43.6%. This is not a big improvement over the approximately 20% accuracy of randomly choosing a genre.

Next, we used MATLAB and the WEKA data mining package to apply Support Vector Machines using the multi-class classification method 1-against-all which breaks the multi-class problem into several binary classification problems. SVMs gave a significant im-

provement in classification accuracy with the linear kernel. This resulted in 77.25% overall classification accuracy. In an attempt to increase the performance, we tried using SVMs with the second power polynomial kernel as well as the RBF kernel. But, it turned out that the linear kernel was the best for this problem; the other two kernels yielded accuracies of 75% and 73%, respectively.

Furthermore, we have experimented with other classifiers hoping that we could find a better classifier than SVMs for this particular problem. In [8] the authors mention that the current state of the art performance in music genre classification is achieved using ensemble classification methods and more specifically Adaboost with a weak classifier Decision Stump. AdaBoost performs large-margin classification by iteratively combining the weighted votes of a collection of weak learners such as Decision Trees or more simply Decision Stumps. We ran Adaboost using 40 boosting iterations and the classification accuracy achieved was 81%; this far surpassed the results we achieved using SVMs.

We wanted to improve this accuracy even more so we decided to explore the boosting techniques. In addition to AdaBoost we also tried the LogitBoost algorithm [11]. LogitBoost uses Newton-stepping with the Hessian, rather than the line search of the generic boosting algorithm. However, as all boosting algorithms are, it is sensitive to overfitting if the number of boosting iterations increases above a specific level. For that reason we have experimented with LogitBoost with the weak learner Decision Stump using between 5 and 50 boosting iterations. Figure 4 shows the classification accuracy we were able to achieve for each of those experiments. Looking at the results, it is clear that when using 40 iterations the accuracy is improved by 2% over AdaBoost. For all individual genres, the accuracy was higher than 76% and the recall was between 70% (recall for Pop genre) and 91.7% with the highest recall to be for the classical music genre. Understandably, for all classifiers, the two most confused genres were Pop and Rock. These are two genres which humans have difficulty defining as well as separating. Classical music was the easiest to separate, which seems natural because it is also the easiest of the five for a person to pick out.

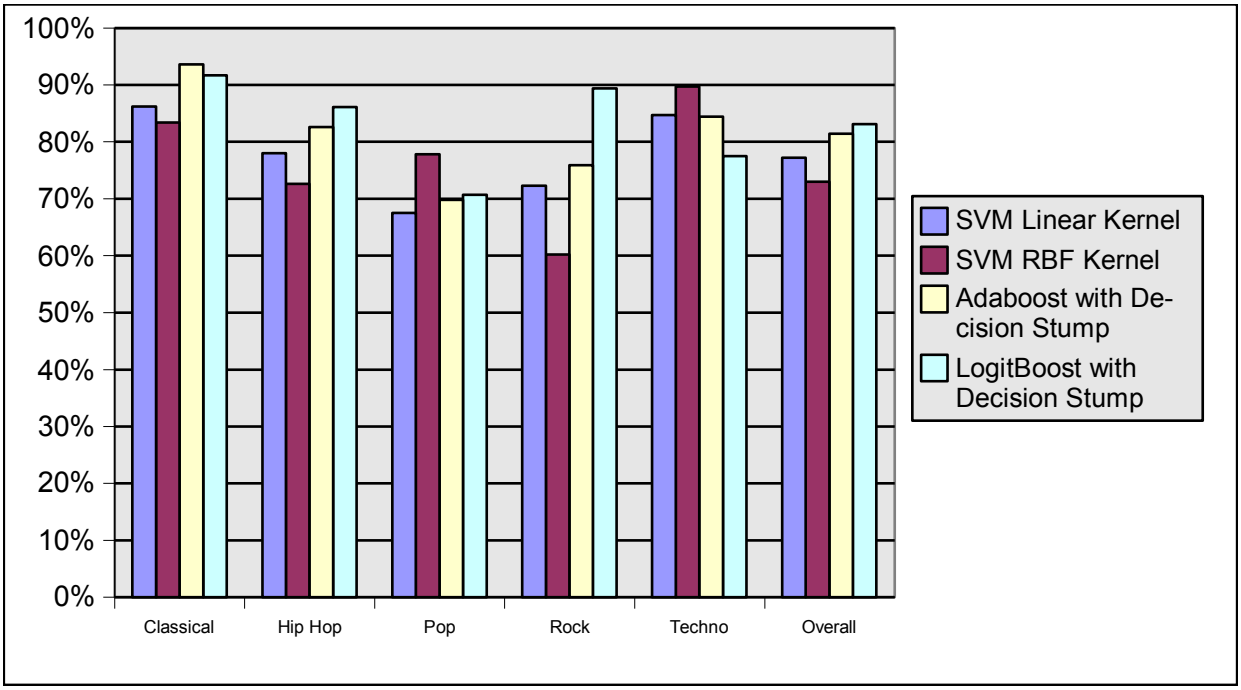

Figure 3: Per class classification accuracies

Conclusion

We have implemented some of the most useful features found in previous research on musical genre classification, as well as a carefully chosen set of new features. The combination of this powerful feature set with a well-chosen classification algorithm allows us to achieve high accuracy genre classification, comparable to that of human accuracy. Both Logitboost and Adaboost are well suited to this problem. Logitboost ultimately performed better because it was very successful in separating the rock genre, which was very intermixed with the other genres. However, Adaboost and the RBV kernel SVM were significantly better at classifying techno songs, which were more isolated from the other genres.

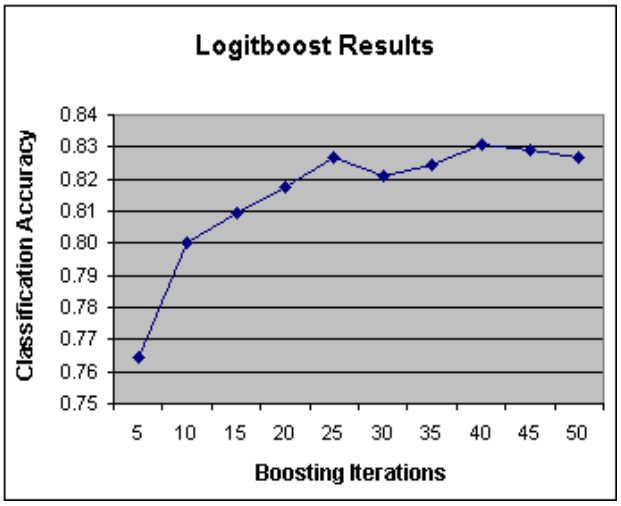

Figure 4: Logitboost accuracy results

Confusion Matrix (40 iteration Logitboost)

	Classical	Hip-Hop	Pop	Rock	Techno
Classical	343	2	13	7	9
Hip-Hop	2	544	48	23	15
Pop	20	57	352	51	18
Rock	15	7	24	554	20
Techno	20	19	25	50	392

(Rows are the actual genres, columns are the classified genres)

References

[1] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, pp. 1331–1334, 1997.

[2] P. Ahrendt and A. Meng, "Music Genre Classifation using the multivariate AR feature integration model," 2005.

[3] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, Vol. 10, No. 5, July 2005.

[4] E. Vincent and X. Rodet, "Instrument Identification in Solo and Ensemble Music Using Independent Subspace Analysis," *ISMR*, 2004.

[5] M. Shan, F. Kuo, M. Chen, "Music Style Mining and Classification by Melody," *Proc. IEEE ICME02*, Lausanne, Switzerland, 2002.

[6] W. Chai and B. Vercoe, "Folk Music Classification Using Hidden Markov Models," *Proc. Of IC-A101*, 2001.

[7] K. Martin and Y. Kim, "Musical Instrument Identification: A Pattern-Recognition Approach," *136th meeting of the Acoustical Society of America*, Oct 13, 1998.

[8] J Bergstra, N Casagrande, D Erhan, D Eck, B Kegl, "Meta-Features and AdaBoost for Music Classification," 2006.

[9] E. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J. Acoust. Soc. Amer.*, vol. 103, no. 1, p. 588, 601, Jan 1998

[10] J. Foote and S. Uchihashi, "The beat spectrum: A new approach to rhythmic analysis," *Proc. Int. Conf. Multimedia Expo.*, 2001

[11] J. Friedman, T. Hastie, R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," Stanford University, 1998.

[12] F. Patkin, "Beat Detection Algorithms," gamedev.net, 2006