Pedro Davalos
CPSC 633-600

May 13, 2009
Final Project Report

Automatic Music Genre Classification

1 – Introduction

In this work, we are presenting our approach to automatic genre classification for music files, or songs, which consists of audio files represented by a time series data, where the goal is to automatically process the files, to establish a genre assignment. Such applications that require automatic genre classification include internet radio stations that play similar songs based on a user specified preference, or for automatic organization of music databases, where large number of songs must be organized and categorized based on style, in addition to the artist information.

This task of genre classification involves processing time series data, but also most significantly involves the classification of subjective data, where some songs might fall into multiple categories, or some categories might be similar to subjective observers. This categorization issue can be explained by a taxonomical analysis of the categories and the multiple genres assigned to music, where for instance some songs might be considered a particular style just based on the artist, even though the particular song diverges from the typical assignment.

Our contribution to the audio processing community, involves an implementation of a successful genre classification algorithm, that as we will describe in detail, consists of a feature extraction process that utilizes Mel-Frequency Cepstral Coefficients (MFCCs) and Spectral and cepstral parameters, and then the data is processed by a Linear Discriminant analysis technique to reduce the dimensionality of our feature vector, such that finally we can then utilize a machine learning technique that consists of a quadratic classifier as an eager learner that processes training data that is pre-labeled by a user trainer, such that the classifier parameters can be learned and estimated for establishing the potential to generalize and classify novel and new unseen test data, or any music file, where based on our testing results, as we will describe in section 5, we are confident that our process will perform adequately for general test cases.

2 – Related Work

The problem of automatically classifying music files to assign a music category or genre has been previously attempted for the multiple benefits that this process would provide. Some of the benefits of automatic genre categorization include an enhanced user experience when listening to music, as some internet radio applications such as Pandora, which plays similar music based on user input. However the audio and signal processing communities have attempted to solve this problem in a similar fashion as speech processing or speaker recognition, although music is a signal with greater complexity than speech, as it includes multiple frequencies, at multiple rhythms, with a mixture of multiple components. Furthermore, ISMIR, the International Symposium for Music Information Retrieval, hosted a competition to solve this problem of automatic genre classification in 2004. This competition provided some guidelines for our motivation for this work in addition to providing the sample training and test data used for this work.

Barbedo and Lopes [3] approached the genre classification problem by also analyzing the music files, but emphasize the taxonomy and the detail categories for genres. Yet Barbedo and Lopes also account in detail their feature extraction and classification technique which is different from ours since they do not use MFCCs or Cepstral information.

Tzanetakis and Cook [4] describe a different technique for music genre classification by focusing heavily on the feature extraction, where they attempt to find features that characterize multiple parts of music such as rhythm and music features. However, they also utilize MFCCs to characterize classical music and vocal intensive music.

Jensen [5] also attempts to solve the music similarity problem by also evaluating the Mel-Frequency Cepstral coefficients (MFCCs) where there is also reference to the same ISMIR competition, and similar results are observed, even though the classification technique is not fully described.

These related attempts to solve the genre classification problem are all similar, yet in this work, we propose a different approach that includes Linear Discriminant Analysis, along with a quadratic classifier as an efficient technique to automatically categorize music. Furthermore, we describe our framework, as well as present successful test results from evaluating our implementation.

3 – Feature Selection

Feature Selection refers to the process of extracting the features from each sample to represent the data. Audio files have the challenge that they represent a time series, where it becomes difficult to find the optimum delta time setting to discretize the data.

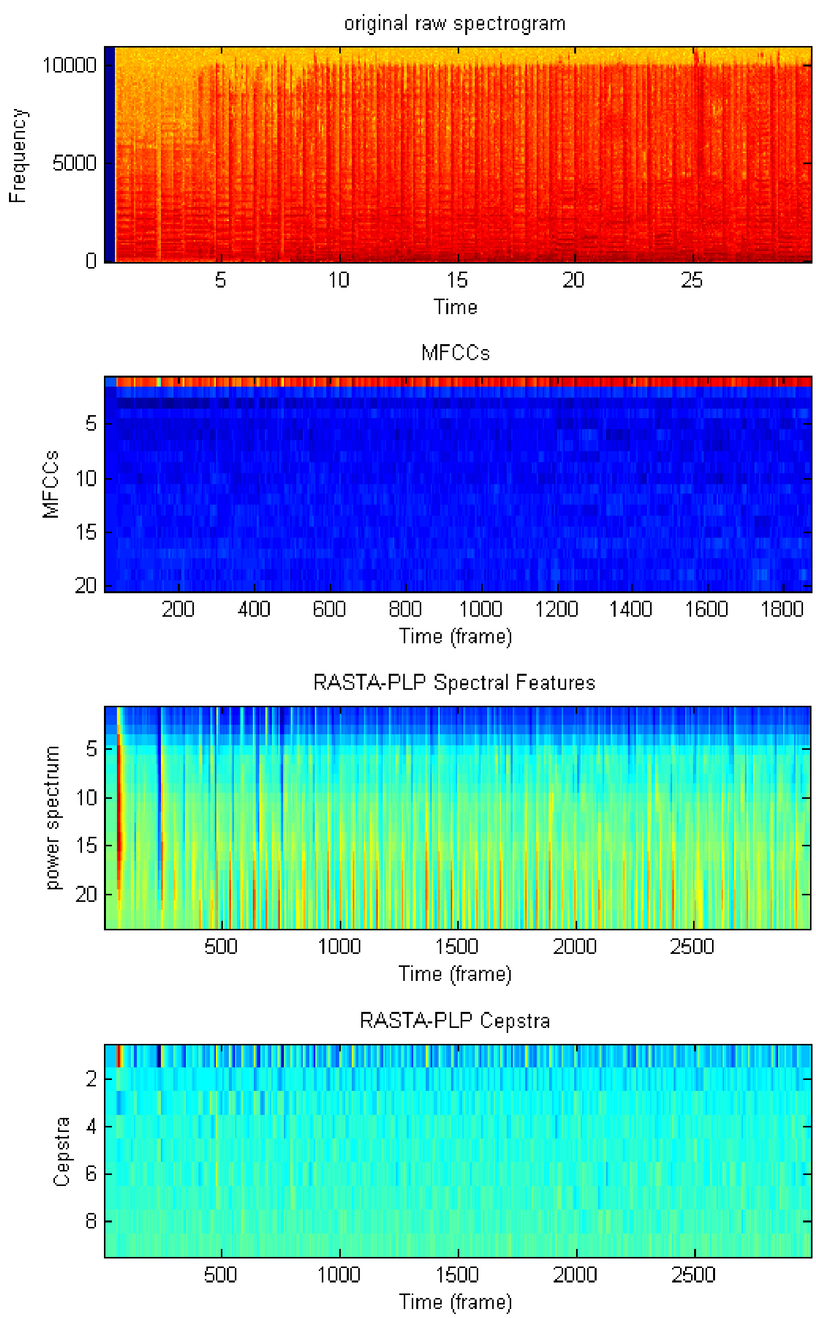
Furthermore, audio data can be represented among orthogonal views, for example speaker recognition should discriminate between speakers regardless of the content, yet speech recognition should discriminate the content regardless of the speaker. And of course, for each application one must select the appropriate features to discriminate accordingly.

However, for our current challenge to discriminate among music genres, we must find appropriate features to represent a broad category of music. Our goal is to learn the characteristics of music genres such that we can then generalize and classify future unseen music files into one of our preset genre classes.

Our approach for feature selection was to represent the data by extracting Mel-Frequency Cepstrum Coefficients (MFCC), as well as rasta-plp spectral features, and cepstral vectors. Yet, because music files are time series, we arbitrarily chose to process only the first 30 seconds of each song, and then we computed the average and the variance for the 30 seconds of each of the three characterizations.

Consequently, our final feature vector for each song, consisted of a 104 dimensional vector composed of the mean and variance of the MFCC's, Spectral, and Cepstral data. Since MFCCs are 20 dimensions, Spectral is 23, and Cepstral is 9, when we combine both the average and the variance, the result is a 104 dimensional vector.

For example, the following figures illustrate a sample 30-second segment of one of the music files, a typical song with the raw spectrogram, and its corresponding MFCC's, along with the Power Spectral data and Cepstral Vectors.



Preprocessing the data to extract the selected features for each song is a required step in the learning process. Our dataset was given by the ISMIR 2004 conference [1], the International Conference on Music

Information Retrieval, where they held a genre classification contest in 2004, and posted sample training and test data along with its corresponding genre labels that include 6 categories of music such as classical, Electronic, jazz_blues, metal_punk, rock_pop, and world.

The dataset consisted of about 700 MP3 music files, however, the downloaded data was corrupted and we were only able to obtain 584 music files from 5 categories, which is sufficient to perform a classification experiment.

4 – Learning and Automatic Classification

Once the data is processed and the 104 dimensional feature vector is extracted for each song, we can perform the learning technique to characterize the data, and to have the ability to generalize and classify new unseen music files based on the training data.

Our learning approach is to divide the 584 music files into randomly chosen training set and a test set, where we only utilize the training set to learn the classification, and then evaluate the remaining test set to analyze the classification performance. This approach can be characterized as an eager learner, where we learn the parameters of our classifier based on user specified and labeled training data. Therefore, once this classifier parameters are trained, the training data can be discarded, and the classification process is an efficient step, that requires minimal effort.

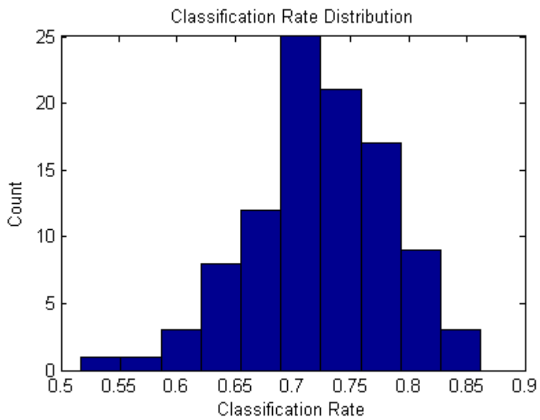
This classification technique used to learn the genre representation, is a quadratic classifier, that finds the mean and the covariance of the data to represent each class.

Furthermore, an additional technique was used to process the data to reduce the dimensionality of the problem. Linear discriminant analysis was performed on the raw 104 dimensional data, to reduce the dimensionality similar to fisher’s linear discriminant process, where the data is projected such that class separation is optimized. The resulting low-dimensional data is a 4-vector that represents each 30-second music file.

5 – Results

Classification performance based on the methods described earlier, are illustrated on the following figure, which shows an average classification rate of about 73% success. The classification rate distribution graph was generated be performing the classification task 100 different times, each time with a different random selection of the training data.

This result can be directly compared to the results posted on by the ISMIR conference that hosted the original genre classification contest, where the 5 published classification rates range between 58% and 84%, which is similar to our results.



6 – Conclusion

In this work, we implemented a classification algorithm for automatic genre classification of music audio files, where we successfully tested on a large database of 584 files, for an optimistic result of about 73% successful rate on average.

This successful results, can be attributed to the feature selection and extraction, where the mean and variance of the MFCC's and the Spectral and Cepstral characterizations have demonstrated to be adequate for genre classification.

Further considerations regarding genre classification also include the subjective nature of music, where perhaps the misclassified songs are borderline between different categories. In other words, the taxonomy of the music categories can be a crucial element in the successful classification.

7 – Future Work

Although Our results for this work on genre classification are successful, and can be compared with the classification rates achieved by competing members of the ISMIR conference, there is still room for improvement. One obvious area where there is potential for improving the classification rates is in the feature selection of the data. The data can be further analyzed to select only features with significant contributions, and also, the 12th order PLP features without RASTA should be considered for evaluation. Furthermore, genre classification should also have room for multiple genre categories for each song. For instance, there may be some blues song that may also fall into the rock division, but this classification is based on a subjective opinion of human listeners, with no objective definite measure to discriminate into one category. However, this experiment followed the guidelines specified by the ISMIR competition where each song was pre-labeled into only one category.

Also, another issue worth further analysis, is our arbitrary selection of the first 30 seconds for each song as the only sample to process for each song, we may want to process the entire song, or also include multiple entries for multiple sections of the file. These issues, along with the configuration of the settings for the MFCC's and the Spectral and Cepstral data, need further considerations and analysis to evaluate the effectiveness of the representations.

8 – References

[1] ISMIR 2004 Genre Contest, http://ismir2004.ismir.net/genre_contest/index.htm,

[2] D. Ellis, "PLP, RASTA, and MFCC Matlab Toolbox", http://labrosa.ee.columbia.edu/matlab/rastamat

[3] Barbedo, J. G. and Lopes, A. 2007. Automatic genre classification of musical signals. EURASIP J. Appl. Signal Process. 2007, 1 (Jan. 2007)

[4] George Tzanetakis and Perry Cook, "Musical Genre Classification of Audio Signals" IEEE Transactions on Speech and Audio Processing, 10(5), July 2002

[5] J. H. Jensen and M. G. Christensen and M. Murthi and S. H. Jensen, "Evaluation of mfcc estimation techniques for music similarity", European Signal Processing Conference, EUSIPCO, 2006

9 – Appendix A: Source Code

Read Data – Feature Extraction

```
clear all; clc; close all;

p=textread('training.txt','%s','delimiter',',');

ind=0;
for i=1:length(p)/6
    i
    pt = ['C:\ML_Final\archive' p{i*6}(2:end)];
    if exist(pt,'file')==2
        ind = ind+1
        [d,sr] = mp3read(pt, [1 30*22050],1,2);
        [mm,aspc] = melfcc(d*3.3752, sr, 'maxfreq', 8000, 'numcep', 20, 'nbands',
                           19, 'fbtype', 'fcmel', 'dcttype', 1, 'usecmp', 1,
                           'wintime', 0.032, 'hoptime',        0.016, 'preemph', 0,
                            'dither', 1);
        [cep1, spec1] = rastaplp(d, sr);
        data(ind,:) = [mean(mm,2)' var(mm,0,2)' mean(cep1,2)' var(cep1,0,2)'
mean(spec1,2)' var(spec1,0,2)'];

        if      strcmp(p{i*6-5},'classical') ==1,   labs(ind)=1;
        elseif  strcmp(p{i*6-5},'electronic')==1,   labs(ind)=2;
        elseif  strcmp(p{i*6-5},'jazz_blues')==1,   labs(ind)=3;
        elseif  strcmp(p{i*6-5},'metal_punk')==1,   labs(ind)=4;
        elseif  strcmp(p{i*6-5},'rock_pop')  ==1,   labs(ind)=5;
        elseif  strcmp(p{i*6-5},'world')     ==1,   labs(ind)=6;
        end
    end
end

save alldata.mat
```

Classification

```
clear; clc; close all;
load alldata

% ------------------------------------------
% Classify

for iij = 1:100
```

```
    pr = randperm(length(data));  % Randomly select training and test data
    mrk = round(length(pr)*.9);      % Use about 90% of data to train
    x1 = data(pr(1:mrk),:);
    clab1 = labs(pr(1:mrk));
    x2 = data(pr(mrk+1:end),:);    % use about 10% of data to test
    clab2 = labs(pr(mrk+1:end));

    [trash2,w,trash]=tamu_lda(x1,clab1');
    x1 = x1*w;  %Project ALL data to LDA
    x2 = x2*w;
    quadraw3 = quadclass(x1,x2,clab1);
    clasrate1(iij) = length(find((quadraw3-clab2)==0))/length(x2);

    test_samples = pr(mrk+1:end);
    badones = find((quadraw3-clab2)~=0);
    badsamples = test_samples(badones);
end

figure; hist(clasrate1);
title('Classification Rate Distribution');
ylabel('Count'); xlabel('Classification Rate');
```

Quadratic Classifier

```
function quadraw=quadclass(x1,x2,clab1)
        clear u x
        for i=1:5 % 9 classes  - Training Parameters
            clear ci
            ci = find(clab1==i);
            u(i,:)=mean(x1(ci,:));
            cm{i}=cov(x1(ci,:))  + 0.1*eye(size(x1,2));  % Regularize Covarience
Matrix
        end

        for i=1:length(x2)  % for each test sample
            x = x2(i,:);
            for g=1:5
                gx(g)= -0.5*(x-u(g,:))* inv(cm{g}) *(x-u(g,:))' -
0.5*log(det(cm{g}));   % cuadratic classifier
            end
            [mx, in]=max(gx);
            quadraw(i)=in;  % Class assignment
        end
```