ECE 366 Honors Section
Fall 2009
Project Description

Introduction: Musical genres are categorical labels created by humans to characterize different types of music. A musical genre is characterized by the common characteristics shared by its members such as instrumentation, rhythmic structure, and harmonic content of music. Until recently musical genre annotation was done manually. Automatic musical genre classification can assist or replace the human user and is valuable for music information retrieval systems. In this project, you will investigate and implement different signal processing based feature extraction methods for musical genre classification. You will implement the feature extraction algorithms in MATLAB and train and test a classifier for genre identification. You will evaluate the performance of the algorithm based on the accuracy of your genre classification.

Background: In this section, I'll provide a brief overview of the most common features [1,2] extracted from music files, in particular the timbral texture features. For the following features, the signal is broken into small, overlapping segments in time referred to as analysis windows. Analysis windows have to be small enough so that the frequency characteristics of the spectrum are stable. Also, in order to capture the long term nature of sound "texture", a longer window, texture window is defined. Usually, an analysis window of 23 ms and a texture window of 1 s is used.

1. **Short-Time Energy**: The energy of the signal in each analysis window is computed and the mean and the variance of the energy over the texture window can be used as features.
2. **Spectral Centroid**: The spectral centroid is defined as the center of gravity of the magnitude spectrum of the Fourier transform. For music analysis, you divide the audio file into overlapping frames of smaller lengths and compute the Fourier transform of each frame, this is also known as the Short-Time Fourier Transform. The spectral centroid is then defined as:
   $C_t = \frac{\sum_{n=1}^{N} nM_t[n]}{\sum_{n=1}^{N} M_t[n]}$, where $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n. The centroid is a measure of spectral shape and higher centroid values correspond to "brighter" timbral textures with more high frequencies. Usually the mean and the variance of the centroid across the different time frames in the texture window are used as features.
3. **Spectral Rolloff**: The spectral rolloff is defined as the frequency R_t below which 85% of the magnitude distribution is concentrated $\sum_{n=1}^{R_t} M_t[n] = 0.85 \sum_{n=1}^{N} M_t[n]$. The mean and the variance of the rolloff across time frames in the texture window are used as features.
4. **Spectral Flux**: The spectral flux is defined as the squared difference between the normalized magnitudes of successive spectral distributions (normalization is done by the total energy in that window) $F_t = \sum_{n=1}^{N} \left(N_t[n] - N_{t-1}[n] \right)^2$ where $N_t[n]$ is the normalized magnitude of the Fourier transform at the current frame t. The spectral flux is a measure

of the amount of local spectral change. The mean and the variance of the flux across time frames in the texture window are used as features.

5. **Time Domain Zero Crossings**: $Z_t = \frac{1}{2}\sum_{n=1}^{N}\left|sign(x[n]) - sign(x[n-1])\right|$. Time domain zero crossings provide a measure of the noisiness of the signal and the mean and the variance of zero crossings across time frames in the texture window are used as features.
6. **Mel-Frequency Cepstral Coefficients (MFCC)**: Mel-frequency cepstral coefficients are perceptually motivated features based on the Fourier transform. After taking the Fourier transform of an analysis window, the magnitude spectrum is passed through a Mel filterbank with varying bandwidth mimicking the human ear, i.e. small bandwidth at low frequencies and large bandwidth at high frequencies. The output energy of each filterbank is log transformed and MFCCs are obtained by taking the Discrete Cosine Transform of the outputs. More specifically, first the frequency is scaled using Mel filterbank H(k,m) and then the logarithm is taken using $X'(m) = \ln\left(\sum_{k=-0}^{N-1}\left|X(k)\right|H(k,m)\right)$ for m=1,2,…,M with M being the number of filterbanks and M<<N (the number of frequency points). The Mel filterbank is a collection of triangular filters defined by center frequencies $f_c(m)$ written as

$$H(k,m) = \begin{cases} 0 & for \quad f(k) < f_c(m-1) \\ \dfrac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)} & for \quad f_c(m-1) \le f(k) < f_c(m) \\ \dfrac{f(k) - f_c(m+1)}{f_c(m) - f_c(m+1)} & for \quad f_c(m) \le f(k) < f_c(m+1) \\ 0 & for \quad f(k) \ge f_c(m+1) \end{cases}$$

   . The center frequencies of the filterbank are computed by approximating the Mel scale with $\phi = 2595\log_{10}\left(\frac{f}{700} + 1\right)$. Then a fixed frequency resolution in the Mel scale is computed, corresponding to a logarithmic scaling of the repetition frequency, using $\Delta\phi = (\phi_{\max} - \phi_{\min}) / (M+1)$, where $\phi_{\max}$ is the largest frequency in the Mel scale. The center frequencies of the filters are computed on the Mel scale as $\phi_c(m) = m\Delta\phi$. To obtain the center frequencies in Hertz use the inverse transform $f_c(m) = 700(10^{\phi_c(m)/2595} - 1)$. Finally, the MFCCs are computed by taking the discrete cosine transform (DCT) of X'(m) $c(l) = \sum_{m=1}^{M} X'(m)\cos(l\frac{\pi}{M}(m - \frac{1}{2}))$ for l=1,2,…M. For music genre identification, usually the first 13 MFCCs are used.
7. **Pitch Detection**: Pitch is defined as the auditory attribute of sound according to which sounds can be ordered from low to high. It is also defined as the fundamental frequency of a harmonic signal. There are different ways to detect the pitch including time domain autocorrelation based methods and frequency domain methods. The simplest algorithm for pitch detection is Harmonic Product Spectrum (HPS). The HPS algorithm measures the maximum coincidence for harmonics in each spectral frame as follows:

$Y(\omega) = \prod_{r=1}^{R} \left| X(\omega r) \right|$ where R is the number of harmonics to be considered and $\hat{Y} = \max_{\omega_i} \{Y(\omega_i)\}$ frequency ω_i is the range of fundamental frequencies. The resulting periodic correlation array, $Y(\omega)$, is searched for a maximum value. Octave errors are a common problem in pitch measurements from HPS. Almost always in these error cases, the pitch is detected one octave too high. To correct for this error, postprocessing should be done with the following rule: If the second peak amplitude *below* initially chosen pitch is approximately 1/2 of the chosen pitch and the ratio of amplitudes is above a threshold (*e.g.*, 0.2 for 5 harmonics), then select the lower octave peak as the pitch for the current frame. Due to noise, frequencies below 50 Hz should not be searched for pitch. There are other algorithms as discussed in [5].

After the different features are extracted, a feature vector of length L representing each audio file will be formed. The feature vectors can be classified using pattern recognition techniques. In this project, since we are more interested in implementing the feature extraction algorithms rather than designing the actual classifiers we will use a simple nearest-neighbor classifier. For each audio file sample, we will classify it as the musical genre that it is closest to. For each music genre, we will first compute the average feature vector representing that group, i.e. $\bar{v}_i = \frac{1}{10}\sum_{k=1}^{10} v_i(k)$ assuming there are 10 samples in each genre class. We will use the take-one-out method and compute the Euclidean distances to each class average and assign each sample to the class that it is closest to, $\frac{1}{L}\sum_{k=1}^{L}(v(k) - \bar{v}_i(k))^2$. The number of correct and incorrect assignments will be counted to find the accuracy rate as $\frac{Number \quad of \quad Correct \quad Assignments}{30}$.

Project Guidelines: You will need to do the following for this project:

1. Data acquisition or collection: You will need to download sample music files and open them in MATLAB. There are many available music databases online where you can download audio files. For example, http://marsyas.sness.net/download/data_sets contains multiple datasets for music genre identification. Of course, you can also use your own MP3 files. You need to choose 10 audio files belonging to three different musical genres such as classical, jazz, rock, blues, reggae, pop, metal, etc. You can use the MATLAB command 'wavread' for reading .wav files. Select a couple of seconds of the audio file for analysis (about 2-3 seconds). In particular, for genre classification I would suggest selecting parts of music where there is no human voice since speech vs. music classification is a different problem.
2. Choice of the feature extraction methods: You need to choose six features (out of the seven provided above) to implement. You can implement the features either using the guidelines given above or can use other implementations, in particular for MFCC and pitch detection there are alternative algorithms that you may want to explore.
3. You will need to divide the audio file into smaller time frames for analysis. You can choose the length of the time frame based on the number of samples in the audio file. Experiment with different number of time frames to see the effect on the extracted features and the classification accuracy. Usually an analysis window of 23 ms and a texture window of 1 s are used. You also need to identify the amount of overlap between the different frames.

4. Evaluation: After you implement your feature extraction and classification algorithms, you need to evaluate them on at least three music genres each with ten audio samples. You will evaluate the classification with respect to the selected features, i.e. first extract only one type of feature and compute the classification accuracy and then combine the features into a larger feature vector and compute the classification accuracy with multiple feature types. You can present your classification results using a table or bar graphs. You will also evaluate how the accuracy changes with respect to the analysis window length, the amount of overlap between the windows (consider at least three different amounts of overlap, i.e. 20%, 50%, 80%), and the parameters used in the feature extraction algorithms such as the number of filters in MFCC extraction.

MATLAB Commands: Some MATLAB commands that can be useful are listed below for your convenience:

- Fft-computes the Fourier transform of the signal
- Fftshift-rearranges the fft values such that they are symmetric with respect to 0 frequency
- Dct- computes the discrete cosine transform
- Sound- can be used to listen to sound files in MATLAB
- Wavread- will read in any .wav file into MATLAB
- Mp3read-read .mp3 files into MATLAB, see the code provided on http://labrosa.ee.columbia.edu/matlab/mp3read.html
- Sum-sums up the values in a vector or matrix
- Mean,Var- compute the mean and variance of vectors/signals
- Abs,angle - Useful for finding the magnitude and the phase of complex functions.

Deliverables: For this project, I expect you to turn in the following:

- A project report: The report should be well-written with the following sections: Abstract, Introduction, Method, Results, Conclusions and References. The abstract should introduce the problem; summarize the approach and major findings in less than 150 words. In the introduction section, you should summarize the problem and the major approaches. This section will require you to do some research on music genre classification methods and synthesize them in your own words. The method section will follow the tasks I outlined above, describing how you chose the different parameters in the design. The results section should summarize your findings in a table format or bar graphs. Finally, you should discuss your findings and comment on how the results can be improved. The report should be typed, 12 point, double-spaced and no more than 10 pages in total. You can use MATLAB code written by others as long as you first test it extensively to make sure that it does what it is supposed to do and cite it.
- MATLAB code that you developed should be added to the project under the Appendix section. This section is not included in the 10-page limit.
- A CD containing the audio files used for genre identification.

Grading: For this project, you will work in groups of 2. Each group needs to meet with me before November 25, 2009. For this meeting, I expect you to do a preliminary research into the different feature extraction methods, choose the six features that you are going to implement, make a choice on which three genres you will be classifying, and download and read music files into MATLAB successfully. This meeting will also give you a chance to ask any implementation related questions that you might have. Each group needs to e-mail me and make an appointment for this meeting (about 20 minutes long).
The grading will be based on the quality of the report, your results and your discussion of the results. I expect all of the groups to implement six feature extraction algorithms including at least

the following components: test the classification algorithm on three music genres with at least 10 audio segments in each genre, quantitative analysis of the effect of the size of the analysis window on the results, quantitative analysis on how discriminative the different features are and the effect of the different design parameters, e.g. fft length, length of the analysis windows, the number of Mel frequency filterbanks etc. You can improve your results and can get extra credit by implementing more advanced classification algorithms and other features such as the ones in [1,2] to improve your identification results and testing your algorithm on a larger dataset of music signals. The final reports and the CDs containing the audio files used are due by December 16, 2009 by 5 p.m.

References:

1. G. Tzanetakis and P. Cook,"Musical Genre Classification of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, July 2002.
2. M. F. McKinney and J. Breebaart, "Features for Audio and Music Classification," Proc. 4th Int. Conf. on Music Information Retrieval, 2003.
3. S. Sigurdsson, K. B. Petersen and T. Lehn-Schioler, "Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music," Proceedings of the International Symposium on Music Information Retrieval, 2006.
4. M. Slaney. Auditory toolbox, version 2. Technical Report #1998-010, Interval Research Corporation, 1998.
5. P. de la Cuadra, A. Master, C.Sapp, "Efficient Pitch Detection Techniques for Interactive Music," International Computer Music Conference, 2001.