Classification of Speech/Music Using Multivariate Features

Shubhanshu Yadav
MS, Computer Engineering, Columbia University
E-Mail: sy2511@columbia.edu

1. INTRODUCTION

Audio Classification plays an important role in understanding the semantic content of the audio data. An automated audio classification system classifying the content into speech, music and silence can be used for many purposes including but not limited to audio indexing, content based audio retrieval and online audio distribution.

Meaningful information can be extracted from digital audio waveforms in order to compare and classify the data efficiently. When such information is extracted, it can be stored as content description in a compact way. These compact descriptors are of great use not only in audio storage and retrieval applications, but also in efficient content-based segmentation, classification, recognition, indexing and browsing of data.

In this paper, we first take a look at the previous work that has been done in this area and analyze the audio features that are typically used to classify the audio data. Next, we take a look at the proposed method which uses a multivariate feature system to give a confidence percentage of the audio content being speech, music.

Keywords

Audio Classification, Bayesian Information Criterion, Feature Extraction, Zero Crossing Rates, Short-Time Energy, Improved Mel-Frequency Cepstral Coefficients (MFCC), Delta Cepstral Energy (DCE), Power Spectrum Deviation (PSDev), Universal Background Model.

2. PREVIOUS WORK

There have been many studies on the audio signal classification using different features and different classification methods and a wide variety of features/methods have been used for the classification into music/speech. The features can be broadly divided into 3 classes according to the domain in which they are calculated, the time domain, the frequency domain or the mixed domain (which includes both time and frequency).

Time-domain features represent the temporal characteristics of the signal. For example, the zero crossing rate (ZCR) [1] ZCR describes the times that an audio waveform crosses the zero axes within a frame. During speech there is an alternation of voiced and unvoiced segments. ZCR is greater during unvoiced segments than voiced segments. So, peaks occur in the evolution of the ZCR during speech. For music, the variations of the ZCR are smoother.

Short-Time Energy (STE) is another such feature. STE is used to measure the strength of audio signals. After calculating the STE, we can then use the percentage of the "low energy frames" [2] as a feature for classification.

Frequency-domain features characterize the spectral envelope of the signal. Some examples of the frequency domain features are, spectral centroid, harmonic coefficients [3] and spectral peak track[4]. The Mel frequency cepstral coefficients parameters (MFCC) are considered as one of the best parameterizations for speech/music discrimination. [5] . The performance of MFCC was improved in [6] and further combined with delta cepstral energy (DCE) and power spectrum deviation (PSDev) [7] to improve the accuracy in classification.

3. Features Used For Audio Classification

Acoustic feature extraction plays an important role in constructing an audio classification system. The aim is to select features which have large between class and small within class discriminative power. Discriminative power of features or feature sets tells how well they can discriminate different classes. Feature selection is usually done by examining the discriminative capability of the features

3.1 Mel Frequency Cepstrum Coefficients

Mel-frequency cepstral coefficients (MFCC) is a parameter used in the discrimination due to the spectral difference between music and speech. A commonly used formula to approximately reflect the relation between the Melfrequency and the physical frequency is given by:-

$$M(f) = 1125 \times log_{10}(1 + \frac{f}{700})$$

In calculating MFCC, the first step is the pre-emphasis. The speech signal s(n) is sent to a high-pass filter. The goal of pre-emphasis is to compensate the high-frequency part that was suppressed during the sound production mechanism of humans. Moreover, it can also amplify the

importance of high-frequency formants. Then, the input speech signal is segmented into frames of 20~30 ms with optional overlap of 1/3~1/2 of the frame size. Usually the frame size (in terms of sample points) is equal to power of two in order to facilitate the use of FFT. If this is not the case, we need to do zero padding to the nearest length of power of two. Each frame then has to be multiplied with a hamming window in order to keep the continuity of the first and the last points in the frame. Spectral analysis shows that different timbres in speech signals corresponds to different energy distribution over frequencies. Therefore we usually perform FFT to obtain the magnitude frequency response of each frame. We multiple the magnitude frequency response by a set of 20 triangular bandpass filters to get the log energy of each triangular bandpass filter. The positions of these filters are equally spaced along the Mel frequency, which is related to the common linear frequency f by the following equation:

mel(f)=1125*ln(1+f/700)

Mel-frequency is proportional to the logarithm of the linear frequency, reflecting similar effects in the human's subjective aural perception.

3.2 Delta MFCC

Features for classification system need to be channel invariant. Instantaneous features, those measures at a single time instant or frame, such as Mel Cepstrum explained above, lack this channel invariance property. Therefore, dynamic features reflecting change over time are required. The delta cepstral serve as a means to obtain dynamic and temporal audio characteristics which can aid speech and music discrimination. The delta cepstral is thus also used in audio indexing systems. The computation of delta MFCC coefficients is given by :-

$$d_t = \frac{\sum_{k=1}^{N} k(c_{t+k} - c_{t-k})}{2\sum_{k=1}^{N} k^2}$$

Where the N represents the delta window size, ct represents the MFCC at time t and dt is the delta coefficient for frame t.

3.3 Power Spectrum Deviation

Speech has greater energy at lower frequencies, however, in the case of music, the higher frequencies also have significant energies. Thus, the energy in each filter of filter bank can also be used for speech and music discrimination. Power Spectrum Deviation (PSDev) is computed as the standard deviation of the filter bank energies in each band. Thus , PSDev can be found using :-

$$P_i = \frac{1}{n-1}\left(\sum_{j=1}^{N}(E_{ij} - E_i)^2\right)$$

Where Ei is the mean energy of all the filters in the ith frame, Pi is the PSDev of the ith frame. N is the numbers of filters in the bank and Eij is the energy in the jth filter of the ith frame.

3.4 RASTA-PLP

Another popular speech feature representation is known as RASTA-PLP, an acronym for Relative Spectral Transform - Perceptual Linear Prediction. PLP was originally proposed by Hynek Hermansky as a way of warping spectra to minimize the differences between speakers while preserving the important speech information [8]. RASTA is a separate technique that applies a band-pass filter to the energy in each frequency subband in order to smooth over short-term noise variations and to remove any constant offset resulting from static spectral coloration in the speech channel.

4. Classification Model

4.1 Universal Background Model (UBM) Adaptation

The paper by Reynolds et al. introduced the concept of UBM adaptation, in which class mixture models are adapted from one background model that represents "all of speech," or a fairly representative cross section of speakers [9]. UBM adaptation distinctly parameterizes the null hypothesis.

Universal background model (UBM) classification is a technique that can be utilized when there is an extremely large amount of available but unlabeled data. Features from this unlabeled data are used to boost the performance of traditional classifiers. Universal background model adaptation is therefore well-suited for this.

For speaker verification, the background model consists of "all speech," whereas in the musical context, the background model consists of "all music". In our case, it will be a combination both the speech and music. Theoretically, the background model should cover a large portion of the feature space.

UBM classification was originally developed for speaker identification. Its creation was motivated by the optimum hypothesis test used for numerous pattern recognition tasks:

$$Hypothesis = \frac{p(Y|H_0)}{p(Y|H_1)} \begin{cases} \geq \theta \ accept \ H_0 \\ < \theta \ reject \ H_0 \end{cases}$$

The log-likelihood version of this test with parameters λ0 for the null hypothesis class and λ1 for the alternative class can be simplified to :-

$$\Lambda(Y) = logp(Y|\lambda_0) - logp(Y|\lambda_1)$$

Although p(Y |λ0) is known (it is simply trained on the target class), p(Y |λ1) is not well defined as it must represent all

alternatives to the null hypothesis. By generalizing the possible space spanned by the null hypothesis alternatives, the universal background model approach allows us to better define p(Y |λ1).

Another advantage to using UBM adaptation as opposed to training an individual Gaussian mixture model (GMM) for each class is that it is assumed that the UBM is already well-trained. GMMs often fit data sub-optimally, but the well-trained parameters of the UBM provide a good starting point for additional models built on top of it.

UBM adaptation and testing consists of several discrete steps. The steps themselves are:

1. Background model creation
2. Background model adaptation to a specific ensemble

1. Background Model Creation

To create the initial background model, a set of features is first extracted from each file and placed in the feature space. A model is then trained on this data to generalize and reduce the parameters of the background model. Most UBM classifiers use Gaussian mixture models (GMMs) to represent the background data. This method can model data with an unknown distribution, and because it is unsupervised, we can train it on our large, unlabeled set of data.

Mathematically, the D-dimensional background model consists of a mixture of K Gaussians represented by the parameter set λ = {w, μ, Σ}:

$$p(x|\lambda) = \sum_{k=1}^{K} w_k p_k(x)$$

$$p_k(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_{k_i}|} exp\left\{-\frac{1}{2}(x-\mu_k)^T(\sum_k^{-1} \quad)(x-\mu_k)\right\}$$

The weights are defined by a hidden variable z that specifies which mixture generated a specific sample, i.e. wk = p(z = k).

The background GMM is trained using the standard expectation-maximization (EM) algorithm. The EM algorithm operates iteratively on the feature set until the convergence criteria are met. Each iteration consists of an E step (which computes the expectation that the data came from the current mixture) and an M step (which maximizes the parameters given the probabilistic alignment of the data):

$$\boldsymbol{E}: P(z_i = k|x_i) \propto w_k N(x_i; \mu_k, \Sigma_k)$$

$$\boldsymbol{M}: w_k = \frac{1}{n}\sum_i P(z_i = k|x_i)$$

$$\mu_k = \frac{\sum_i P(z_i = k|x_i)\, x_i}{P(z_i = k|x_i)}$$

$$\boldsymbol{\Sigma_k} = \frac{\sum_i P(z_i = k|x_i)\,(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i P(z_i = k|x_i)}$$

Parameter selection, such as the number of Gaussians, is application-specific and can be determined by cross validation on the training data. Because the iterative nature of the EM algorithm is sensitive to starting conditions, it is possible for different models to be developed on the same training set. Furthermore, mixtures may have too much overlap, or convergence may not be reached at all. Several methods exist to combat these issues. Many different GMMs trained on the same data may be pruned to generate a good overall fit that has not been affected by outliers. For the overlap problem, the initial means of each mixture can be set to a random data point. Number of iterations can also be increased in order to guarantee convergence.
The best performance in the UBM case has been empirically observed [9] to use a diagonal covariance matrix for each mixture.

Once the GMM has been trained on the background data, we save the parameters in the UBM density:

$$pUBM(x|\lambda_{UBM}) = \sum_{k=1}^{K} w_k p_k(x)$$

2. UBM Adaptation

Given the universal background model and some additional data, we then adapt the UBM to fit the hypothesis class. The steps taken to adapt the model are similar to the EM algorithm, but use a form of Bayesian inferencing for the maximization step. Essentially, a blend of the parameters of the UBM and those of a GMM trained on the hypothesis class results in the final adapted model.

To begin, the same features used for the UBM are extracted from the new data X, with X = {x1, . . . , xT}. Using the well-trained model, we then compute the probabilistic assignment of each new training point in the background model:

$$\Pr(k|x_t) = \frac{w_k p_k(x_t)}{\sum_{j=1}^{K} w_j p_j(x_t)}$$

Next, sufficient statistics for the adapted model's parameters are computed:

$$n_k = \sum_{t=1}^{T} \Pr(k|x_t)$$

$$E_k(x) = \frac{1}{n_k}\sum_{t=1}^{T} \Pr(k|x_t)\, x_t$$

$$E_k(x^2) = \frac{1}{n_k}\sum_{t=1}^{T} \Pr(k|x_t)\, x_t^2$$

A blending coefficient α, used to set the sensitivity of the adaptation, is also introduced. If an existing cluster in the UBM has a low probabilistic alignment with the new training data, it is better to use the original, well-trained parameters instead of using the new, possibly undertrained parameters. Therefore, a relevance factor r is set to force mixtures with a low probabilistic count to retain their original UBM parameters. r is used to calculated the blending coefficient α, and α is used to compute the adapted model parameters:

$$\alpha = \frac{n_k}{n_k + r}$$

$$w_k = \frac{1}{\gamma}\left[\frac{\alpha\eta_k}{T} + (1-\alpha)w_k\right]$$

$$\mu_k = \alpha E_k(x) + (1-\alpha)\mu_k$$

$$\sigma_k^2 = \alpha E_k(x)^2 + (1-\alpha)(\sigma_k^2 + \mu_k^2) - \mu_k^2$$

As the probabilistic count of a cluster approaches 0, i.e. nk → 0, α → 0 and the original UBM parameters are retained. The parameter r can be derived with cross validation. The best empirical results were achieved when r was equal to about 10% of the number of new training points.

In addition, it is not required that all parameters are updated. Better results can be obtained by not updating the variance of the original UBM clusters.

An advantage of UBM adaptation over typical GMM classification is its resistance to overfitting. As the number of components in a GMM approaches the number of points in a training set, the probability that the model is overfitting increases. Due to the relevance factor and the fact that not all UBM components are updated during adaptation, this allows us to use a relatively small training set and a larger number of mixtures with less risk of overfitting. This provides an advantage, as increasing the number of mixtures can increase the overall accuracy of a mixture model.

4.2 Bayesian Information Criterion

BIC is a likelihood criterion penalized by the model complexity: the number of parameters in the model[10]. In detail, let $X_i = \{x_i : i = 1, \dots N\}$ be the data set we are modeling; let $M = \{M_i : i = 1, \dots, K\}$ be the candidates of desired parametric models. Assuming we maximize the likelihood function separately for each model *M*, obtaining, say L(X,M). Denote #(M) as the number of parameters in the model *M*. The BIC criterion is defined as:

$$BIC(M) = \log L(X, M) - \lambda\frac{1}{2}\#(M) \times \log(N)$$

where the penalty weight $\lambda = 1$. The BIC procedure is to choose the model for which the BIC criterion is maximized.

This procedure can be derived as a large-sample version of Bayes procedures for the case of independent, identically distributed observations and linear models.

Change detection is formulated as a hypothesis testing problem. We assume that there are two neighboring chunks X and Y around time cj and the problem is to decide whether or not a speaker change point exists on cj. Let Z = X ∪ Y.

Under H0 there is no speaker change at time cj . The maximum likelihood (ML) principle is used to estimate the parameters of the chunk Z that is modelled by a GMM of two components. Let us denote the GMM parameters estimated by the expectation maximization (EM) algorithm as θz. The log likelihood L0 is calculated as:-

$$L_0 = \sum_{i=1}^{N_x} logp(x_i|\theta_z) + \sum_{i=1}^{N_y} logp(y_i|\theta_z)$$

where Nx and Ny are the total number of samples in chunks X and Y , respectively.

Under H1 there is a speaker change at time cj . The chunks

X and Y are modelled by multivariate Gaussian densities whose parameters are denoted by θx and θy. Then, the log likelihood L1 is given by:-

$$L_1 = \sum_{i=1}^{N_x} logp(x_i|\theta_x) + \sum_{i=1}^{N_y} logp(y_i|\theta_y)$$

The dissimilarity is estimated by:-

$$d = L_1 - L_0 - \frac{\lambda}{2}.\Delta K.\log(N_x + N_y)$$

where λ is the penalty factor (ideally 1.0) tuned according to data and ΔK is the number of the model parameters. If d > 0 then a local maximum is found and time cj is considered to be a speaker change point. Otherwise, there is no change point at time cj .

The selection of the appropriate features is of great importance since the accurate description of the audio signal is vital. We utilize the mel cepstrum coefficients (MFCCs) and Delta Cepstral Coefficients to do that.

Every speaker is represented with a multivariate Gaussian probability density function. So for every speaker we keep the mean vector μ and the covariance matrix Σ that are automatically updated when more data are available. Utilizing the fact that the chunks are becoming larger, we employ a constant updating of the speaker models.

5. Implementation

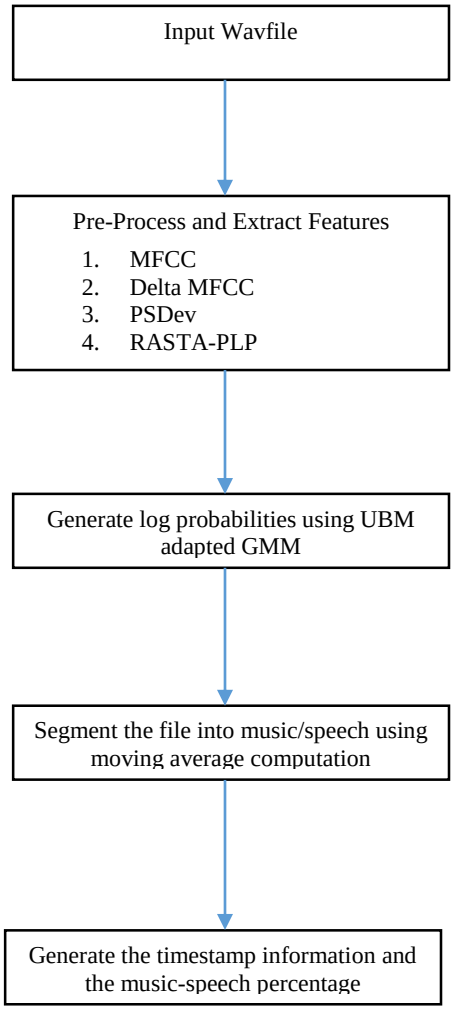

Figure : Block Diagram for speech-music classification

5.1 Signal Pre-processing

Audio signal has to be preprocessed before extracting features. There is no added information in the difference of two channels that can be used for classification or segmentation. Therefore it is desirable to have a mono signal to simplify later processes. The algorithm checks the number of channels of the audio. If the signal has more than one channel, it is mixed down to mono.

The amplitude of the signal is then normalized to the maximum amplitude of the whole file to remove any effects the overall amplitude level might have on the feature extraction.

5.2 Feature Extraction

4 sets of features are extracted from each frame of the audio by using the feature extraction techniques. Here, the low level features are Mel Frequency Cepstrun Coefficients (MFCC), Delta MFCC, Power Spectrum Deviation and Relative Spectral Transform - Perceptual Linear Prediction (RASTA-PLP). These 4 sets of feature values will be calculated for each frame of the given wavfile. A 2D matrix is generated due to this, where each row represents each frame of the given wavfile and the columns represent the features. The number of MFCC coefficients calculated are 13.

5.3 Classification

5.3.1 Creation of UBM Adapted GMM

5.3.1.1 Dataset Used

The data used to train the background model must be sizable, representative of the data that will be seen in each class. So the data required for the UBM creation should be audio files containing only speech, files with only music and files with both speech and music.

Therefore, I used the entire training dataset as the data for UBM creation. The training set consists of 4 types of files – speech, music, music+speech, other.

5.3.1.2 Number of Mixtures

It is a rule of thumb[11] that the number of mixtures in a Gaussian mixture model should depend on the number of datapoints.

Hence I am using a single mixture for the GMMs.

5.3.1.3 GMM Covariance

Similar amount of accuracy can be achieved with fewer mixtures, full covariance or with more mixtures with diagonal covariance[12].

Since I am using a single mixture, I have decided to use full covariance.

5.3.1.4 UBM Adapting only Means

When adapting the UBM to fit a particular class, trying to adapt covariance leads to incorrect averages between the UBM and Class data. It is possible to obtain the same results as adapting covariance by having a low relevance factor[12].

So I based on that, I am only adapting the means(centers).

5.3.2 Classification of the given audio file

The given audio file is then classified using the UBM adapted GMM. The log probabilities for music and speech are generated for each frame of the given audio file using this approach. After this, we make use of moving window

computation to segment the audio file into music and speech. Typically, an empirically determined number of frames are taken in the beginning. Then, the log probabilities of both speech and music of those frames are summed and then checked for a higher value. This gives a starting value. We then begin moving this frame window and look for crossover points at which the summed probability of one (speech or music) becomes greater than the other. Using this way, we segment the whole file into music and speech. The timestamp information generation is also incorporated into this step as computing them separately will become computationally costly. Therefore, at the end of this step, we get an information of the percentage of speech and music, as well as the timestamp information describing which portions of the given audio file is speech and music.

5.3.3 Classification using BIC

In another approach taken for classification, Bayesian Information Criterion is used to detect the change points in the given audio file. In a 2 model classification problem such as ours, these change points can be taken to mean as the points where the segment of one class ends and the other begins. These change points are then used to generate the segments of the given audio file and the process described above is used to classify the segments.

6. Results and Discussion

For classification, the audio files other than the files used for training are tested. The extracted feature vector is used to classify whether the audio is speech or music. The UBM adapted GMM model is created and frame based classification is then performed on the audio files. For generating the UBM adapted GMM model, a total of about 190 files were used. For testing, 60 files, each of 15 seconds, were used overall. Out of those 60 files, 20 files were of speech, 20 files were of music with no vocals and 20 files were of music with vocals.

The results that were obtained are as follows :-

File Type	Classification Percentage
Speech	81.82%
Music with no vocals	66%
Music with vocals	91.8%

The detailed results of each test file can be found in "obtained_results" folder.

Classification with BIC:-

In classification with BIC, all the process up until the creation of UBM adapted GMM are the same. After that, a given test audio file is divided into segments based on the change points generated by the BIC process. In a 2 model classification problem such as ours, these change points can be taken to mean as the points where the segment of one class ends and the other begins. Therefore, for each segment, we generate the log probabilities of its frames and then compare the probabilities to classify the frames as music or speech. This is then used to classify a complete segment as either speech or music. The percentage of speech and music present in that given audio file is then calculated from the number of segments being speech or music. I feel that although I have implemented the BIC to generate the change points correctly, I have not implemented the classification process after that correctly.

All the detailed results of each test file can be found in "obtained_results" folder. This folder contains the result from both the implementations.

7 Conclusion

In this project, we have used 2 methods to classify the audio file into speech and music.

From the results, we can see that the given implementation works very well for speech files and files containing music with vocals, but doesn't work very well for just music files. Also, the moving average computation was tried with 3 frames and 5 frames. Since the better results were obtained when taking the 5 frames window length, it is used in the implementation.

Also, the accuracy obtained from BIC based classification closely tracks the accuracy obtained from the other implementation.

I. REFERENCES

[1] C. Panagiotakis, G. Tziritaz, "A Speech/Music Discriminator Based on RMS and Zero-Crossings," *IEEE Trans. on Multimedia*, vol.7 (1), pp. 155-166, 2005.

[2] W.Q. Wang, W. Gao, D.W. Ying, "A Fast and Robust Speech/Music Discrimination Approach," *Information, Communications and Signal Processing, 2003*, vol.3, pp. 1325-1329, 2003.

[3] E. Scheirer, and M. Slaney, "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator," *Int. Conf. Acoustic, Speech, and Signal Processing (ICASSP'97)*, Munich, Germany, vol. 2, pp. 1331-1334, 1997.

[4] Taniguchi, T., Tohyama, M., Katsuhiko, S., 2008. Detection of speech and music based on spectral tracking. In: Speech Communication, vol. 50, pp. 547–563.

[5] Sannella, Mubarak, O., Ambikairajah, E., and Epps, J., "Analysis of an MFCC-based Audio Indexing system for efficient coding of multimedia sources" in Proc. The Eighth International Symposium on Signal Processing and its

applications, August 28 - 31, 2005 (Australia) Pages: 619 - 622

[6] Huiyu Zhou, Abdul Sadka and Richard M. Jiang, "Feature Extraction for Music and Speech Discrimination", *Content-Based Multimedia Indexing*. pp. 170-173, 2008

[7] Omer Mohsin Mubarak, Eliathamby Ambikairajah and Julien Epps, "Novel Features for Effective Speech and Music Discrimination", *IEEE International Conference on Engineering of Intelligent Systems*,2006,pp. 1-5

[8] Hynek Hermansky , Nelson Morgan , Aruna Bayya , Phil Kohn, "Rasta-Plp Speech Analysis", 1991

[9] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted Gaussian mixture models. Digital Signal Processing, 10(1-3):19–41, 2000.

[10] Kotti, M, Benetos, E Kotropoulos, C Martins, LG, "Speaker change detection using BIC: a comparison on two datasets", 2nd International Symposium on Communications, Control and Signal Processing, 2013.

[11] Wikipedia – Determining the number of clusters in a data set,http://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set.

[12] Brian Dolhansky, Musical Ensemble Classification Using Universal Background Model Adaptation and the Million Song Dataset

[13] Alain Tritschler,Ramesh Gopinath "Improved Speaker Segmentation and Segmentation Clustering Using Bayesian Information Criterion", IBM T. J. Watson Research Center