Volume 3, Issue 9, September 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Speech/Music Classification using SVM and GMM

R.Thiruvengatanadhan*
Department of Computer Science and Engineering
Annamalai University
Annamalainagar, Tamilnadu, India

P.Dhanalakshmi
Department of Computer Science and Engineering
Annamalai University
Annamalainagar, Tamilnadu, India

Abstract— Today, digital audio applications are part of our everyday lives. Automatic audio classification is very useful in audio indexing; content based audio retrieval and online audio distribution. The accuracy of the classification relies on the strength of the features and classification scheme. In this work both, time domain and frequency domain features are extracted from the input signal. Time domain features are Zero Crossing Rate (ZCR) and Short Time Energy (STE). Frequency domain features are spectral centroid, spectral flux, spectral entropy and spectral roll-off. After feature extraction, classification is carried out, using Support Vector Machine (SVM) and Gaussian Mixture Model (GMM). GMM is a classical technique taken as reference for comparing the performance of SVM in terms of accuracy and execution time. The proposed feature extraction and classification models results in better accuracy in speech/music classification.

Keywords— Feature Extraction, Time domain features, Frequency domain features, Classification, Support Vector Machine, Gaussian Mixture Model.

I. INTRODUCTION

The term audio is used to indicate all kinds of audio signals, such as speech, music as well as more general sound signals and their combinations. Multimedia databases or file systems can easily have thousands of audio recordings. However, the audio is usually treated as an opaque collection of bytes with only the most primitive fields attached; namely, file format, name, sampling rate, etc. 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.

The need to automatically classify, to which class an audio sound belongs, makes audio classification and categorization an emerging and important research area [10]. During the recent years, there have been many studies on automatic audio classification using several features and techniques. A data descriptor is often called a feature vector and the process for extracting such feature vectors from audio is called audio feature extraction. Usually a variety of more or less complex descriptions can be extracted to feature one piece of audio data. The efficiency of a particular feature used for comparison and classification depends greatly on the application, the extraction process and the richness of the description itself. Digital analysis may discriminate whether an audio file contains speech, music or other audio entities. A method is proposed in [6] for speech/music discrimination based on root mean square and zero crossings. The quality of a digital audio recording depends heavily on two factors: the sample rate and the sample format or bit depth. Increasing the sample rate or the number of bits in each sample increases the quality of the recording, but also increases the amount of space used by audio files on a computer or disk. Sample rates are measured in hertz (Hz), or cycles per second. This value simply represents the number of samples captured per second in order to represent the waveform; the more samples per second, the higher the resolution, and thus the more precise the measurement is of the waveform. The human ear is sensitive to sound patterns with frequencies between approximately 20 Hz and 20,000 Hz. Sounds outside that range are essentially inaudible.

II. ACOUSTIC FEATURES

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 well they can discriminate different classes. Feature selection is usually done by examining the discriminative capability of the features [5][3].

A. Time Domain Features

1.) Zero Crossing Rate

In case of discrete time signals, a zero crossing is said to occur if there is a sign difference between successive samples. The zero crossing rates (ZCR) are a simple measure of the frequency content of a signal. For narrow band signals, the average zero crossing rate gives a reasonable way to estimate the frequency content of the signal. But for a

broad band signal such as speech, it is much less accurate. However, by using a representation based on the short time average zero crossing rate, rough estimates of spectral properties can be obtained [4].In this expression, each pair of samples is checked to determine where zero crossings occur and then the average is computed over *N* consecutive samples.

$$Zm = \sum_n |sgn\,[x(n)] - sgn\,[x(n-1)]|w(m-n) \qquad (1)$$

Where the *sgn* function is

$$sgn[x(m)] = \begin{cases} 1, x(m) \geq 0 \\ -1, x(m) < 0 \end{cases} \qquad (2)$$

And *x(n)* is the time domain signal for frame *m*.

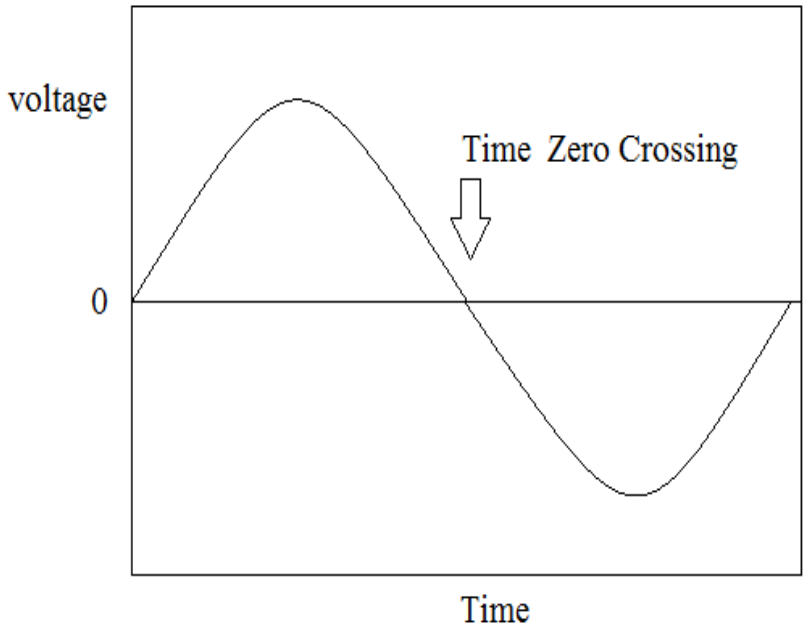

Fig. 1 Zero Crossing Rate

2.) *Short Time Energy*

Short Time Energy (STE) is used indifferent audio classification problems. In speech signals, it provides a basis for distinguishing voiced speech segments from unvoiced ones. In case of very high quality speech, the short term energy features are used to distinguish speech from silence. The energy *E* of a discrete time signal *x(n)* is defined by the expression.

$$E = \sum_{n=-\infty}^{\infty} x^2\,(n) \qquad (3)$$

The amplitude of an audio signal varies with time. A convenient presentation that reflects these amplitude variations is the short time energy of the signal. In general, the short time energy is defined as follows.

$$E_m = \sum_n [x(n)w(m-n)]^2 \qquad (4)$$

The above expression can be rewritten as

$$E_m = \sum_n x\,(n)^2 h(m-n) \qquad (5)$$

Where *h(m) = w2(m).* The term *h(m)* is interpreted as the impulse response of a linear filter. The choice of the impulse response, *h(n)* determines the nature of the short time energy representation. Short time energy of the audible sound is in general significantly higher than that of silence segments. In some of the systems, the Root Mean Square (RMS) of the amplitude is used as a feature for segmentation. It can be used as the measurement to distinguish audible sounds from silence when the SNR (signal to noise ratio)is high and its change pattern over time may reveal the rhythm and periodicity properties of sound. These are the major reasons for using STE in segmenting audio streams of various sounds and categories [7].

B. Frequency Domain Features

1.) Spectral Centroid

The spectral centroid is a measure used in digital signal processing to characterize a spectrum. It indicates where the "center of mass" of the spectrum is. Perceptually, it has a robust connection with the impression of "brightness" of a sound. Itis calculated as the weighted mean of the frequencies present in the signal, which is determined using a Fourier transform, with their magnitudes as the weights

$$centroid = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)} \qquad (6)$$

Where *x(n)* represents the weighted frequency value, or magnitude, of bin number *n*, and *f(n)* represents the center frequency of that bin. Some people use "spectral centroid" to refer to the median of thespectrum. This is a different statistic, the difference being essentially the same as the difference between un weight median and mean statistics. Since both are measures of central tendency, in some situations they will exhibit some similarity of behavior. But since typical audio spectra are not normally distributed, the two measures will often give strongly different values. Grey and Gordonin 1978 found the mean a better fit than the median. Because the spectral centroid is a good predictor of the "brightness" of a sound, it is widely used in digital audio and music processing as an automatic measure of musical timbre.

2.) Spectral Flux

Spectrum flux (SF) is defined as the average variation value of spectrum between two adjacent frames in a given clip. In general, speech signals are composed of alternating voiced sounds and unvoiced sounds in the syllable rate, while music signals do not have this kind of structure. Hence, for speech signal, its spectrum flux will be in general greater than that of music. The spectrum flux of environmental sounds is among the highest, and changes more dramatically than those of speech and music. This feature is especially useful for discriminating some strong periodicity environment sounds such as tone signal, from music signals. Spectrum flux is a good feature to discriminate among speech, environment sound and music.

$$SF = \frac{1}{(N-1)(k-1)} \sum_{n=1}^{N-1} \sum_{k=1}^{k-1} [logA(n,k) - logA(n-1,k)]^2 \qquad (7)$$

Where *A(n,k)* is the discrete Fourier transform of the *nth* frame of input signal.

$$A(n,k) = \left| \sum_{m=-\infty}^{\infty} x(m)w(nL-m)e^{j\frac{2\pi}{L}Km} \right| \qquad (8)$$

x(m) is the original audio data, *w(m)* is the window function, *L* is the window length, *K* is the order of Discrete Fourier Transform (DFT), and N is the total number of frames. This feature cans be used in both speech/music classification and pure speech/speech over background sound classification. In a variation of the feature i.e. Variance of the spectrum flux and variance of ZCR are used.

3.) Spectral Roll off

As the Spectral Centroid, the Spectral Roll off is also a representation of the spectral shape of a sound and they are strongly correlated. It's defined as thefrequency where 85% of the energy in the spectrum is below that frequency. If *K* is the bin that full fills

$$\sum_{n=0}^{k} x(n) = 0.85 \sum_{n=0}^{N-1} x(n) \qquad (9)$$

Then the Spectral Roll off frequency is *f(K)*, where *x(n)* represents the magnitude of bin number *n*, and *f(n)* represents the center frequency of that bin.

4.) Spectral Entropy

The spectral entropy is the quantitative measure of the spectral disorder. The entropy has been used to detect silence and voiced region of speech in voice activity detection. The discriminatory property of this feature gives rise to its use in speech recognition. The entropy can be used to capture the formants or the peakness of a distribution. Formants and their locations have been considered to be important for speech tracking[9].

$$E = -\sum_{i=0}^{L-1} n_i * \log_2(ni) \qquad (10)$$

III. CLASSIFICATION MODEL

A.) Support Vector Machine

SVM is a statistic machine learning technique that has been successfully applied in the pattern recognition area [11], [13] and, is based on the principle of structural risk minimization (SRM) [12]. SVM constructs a linear model to estimate the decision function using non-linear class boundaries based on support vectors. If the data are linearly separated, SVM trains linear machines for an optimal hyper plane that separates the data without error and into the maximum distance

between the hyper plane and the closest training points. The training points that are closest to the optimal separating hyper plane are called support vectors.

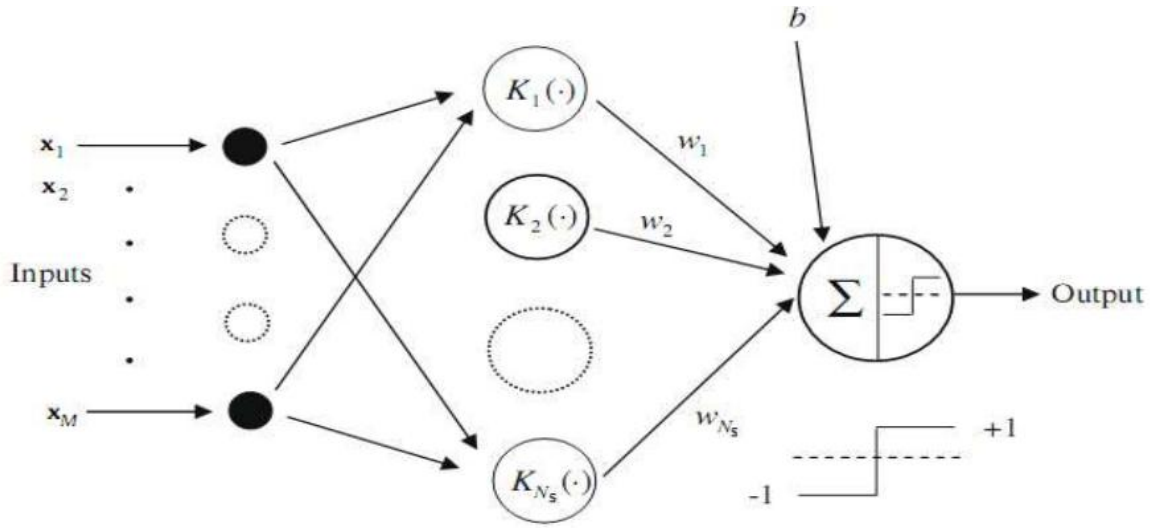

Fig. 2: Architecture of the SVM (Ns is the number of support vectors).

Fig. 2 shows the architecture of the SVM. SVM maps the input patterns into a higher dimensional feature space through some nonlinear mapping chosen a priori. A linear decision surface is then constructed in this high dimensional feature space. Thus, SVM is a linear classifier in the parameter space, but it becomes a non-linear classifier as a result of the non-linear mapping of the space of the input patterns into the high dimensional feature space.

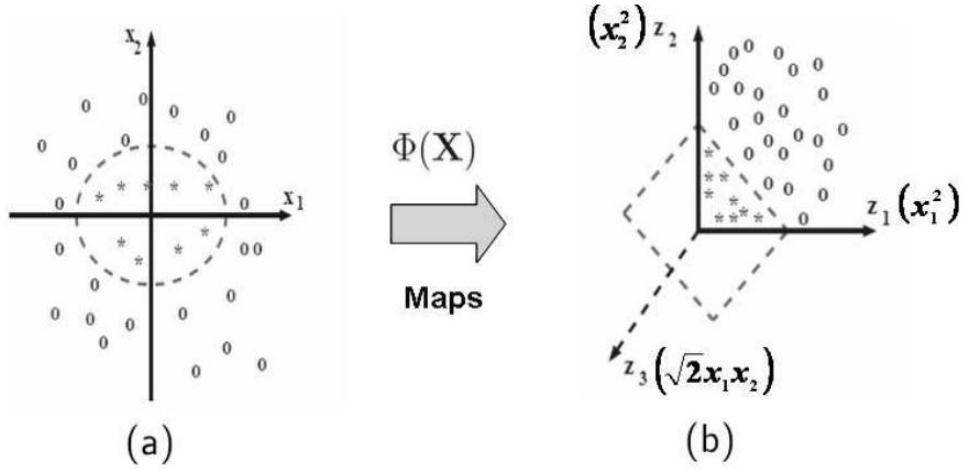

Fig. 3: An example for SVM kernel function Φ(x) maps 2-dimensional input space to higher 3-dimensional feature space. (a) Nonlinear problem. (b) Linear problem.

For linearly separable data, SVM finds a separating hyper plane which separates the data with the largest margin. For linearly inseparable data, it maps the data in the input space into a high dimension space $x \in R^I \rightarrow \emptyset(x) \in R^H$ with kernel function$\emptyset(x)$, to find the separating hyperplane. An example for SVM kernel function $\emptyset(x)$ maps 2-Dimensional input space to higher 3-Dimensional feature space as shown in Fig. 3. SVM was originally developed for two class classification problems. The N class classification problem can be solved using N SVMs. Each SVM separates a single class from all the remaining classes.

SVM generally applies to linear boundaries. In the case where a linear boundary is in appropriate SVM can map the input vector into a high dimensional feature space. By choosing a non-linear mapping, the SVM constructs an optimal separating hyper plane in this higher dimensional space, as shown in Fig. 3. The function K is defined as the kernel function for generating the inner products to construct machines with different types of non-linear decision surfaces in the input space.

$$K(x, x_i) = \emptyset(x).\emptyset(x)_i \qquad (11)$$

The kernel function may be any of the symmetric functions that satisfy the Mercer's conditions (Courant and Hilbert, 1953). There are several SVM kernel functions are

1.) Gaussian Kernel

The Gaussian kernel is an example of radial basis function kernel.

$$K(x, x_i) = \exp\left(-\frac{|x-x_i|^2}{2\sigma^2}\right) \qquad (12)$$

Alternatively, it could also be implemented using

$$K(x, x_i) = \exp(-\gamma|x - x_i|^2) \qquad (13)$$

The adjustable parameter *sigma* plays a major role in the performance of the kernel and should be carefully tuned to the problem at hand. If overestimated, the exponential will behave almost linearly and the higher-dimensional projection will start to lose its non-linear power. In the other hand, if underestimated, the function will lack regularization and the decision boundary will be highly sensitive to noise in training data.

2.) Sigmoidal kernel

Sigmoidal kernel functions which aren't strictly positive definite also have been shown to perform very well in practice. Despite its wide use, it is not positive semi-definite for certain values of its parameters.

$$\tanh(\beta_0 x^T x_i + \beta_1) \qquad (14)$$

where x_i is support vectors, $\beta 0$, $\beta 1$ are constant values.

3.) Polynomial Kernel

The Polynomial kernel is a non-stationary kernel. Polynomial kernels are well suited for problems where all the training data is normalized.

$$K(x, x_i) = (ax^T x_i + c)^d \qquad (15)$$

Adjustable parameters are the slope *alpha*, the constant term *c* and the polynomial degree *d*.

The dimension of the feature space vector $\emptyset(x)$ for the polynomial kernel of degree *p* and for the input pattern dimension of *d* is given by

$$\frac{(p+d)!}{p!\,d!} \qquad (16)$$

For sigmoidal kernel and Gaussian kernel, the dimension of feature space vectors is shown to be infinite. Finding a suitable kernel for a given task is an open research problem. Given a set of audio corresponding to N categories for training, N SVMs are trained. Each SVM is trained to distinguish between one category and all other categories in the training set. During testing, the class label l of an audio x can be determined using (17)

$$l = \begin{cases} n, & \text{if } d_n(\boldsymbol{x}) + t > 0 \\ 0, & \text{if } d_n(\boldsymbol{x}) + t \leq 0 \end{cases} \qquad (17)$$

where $d_n(x) = \max\{d_i(x)\}_{i=1}^N$, and $d_i(x)$ is the distance from x to the SVM hyper plane corresponding to category i. The classification threshold is t, and the class label l = 0 stands for unknown.

B.) Gaussian Mixture Model

The Gaussian mixture model (GMM) is used in classifying different audio classes. The Gaussian classifier is an example of a parametric classifier. It is an intuitive approach when the model consists of several Gaussian components, which can be seen to model acoustic features. In classification, each class is represented by a GMM and refers to its model. Once the GMM is trained, it can be used to predict which class a new sample probably belongs to [14].

The probability distribution of feature vectors is modeled by parametric or non-parametric methods. Models which assume the shape of probability density function are termed parametric. In non-parametric modeling, minimal or no assumptions are made regarding the probability distribution of feature vectors. The potential of Gaussian mixture models to represent an underlying set of acoustic classes by individual Gaussian components, in which the spectral shape of the acoustic class is parameterized by the mean vector and the covariance matrix, is significant. Also, these models have the ability to form a smooth approximation to the arbitrarily-shaped observation densities in the absence of other information [15]. With Gaussian mixture models, each sound is modeled as a mixture of several Gaussian clusters in the feature space.

The basis for using GMM is that the distribution of feature vectors extracted from a class can be modeled by a mixture of Gaussian densities as shown in Fig. 4.

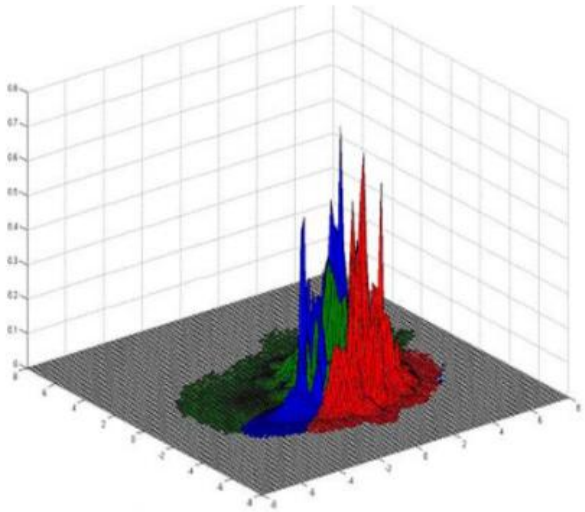

Fig. 4. Gaussian mixture models.

For a D dimensional feature vector x, the mixture density function for category s is defined as

$$p(\frac{x}{\lambda^s}) = \sum_{i=1}^{M} \alpha_i^s f_i^s(x) \qquad (18)$$

The mixture density function is a weighted linear combination of m component uni-modal Gaussian densities $f_i^s(.)$. Each Gaussian density function $f_i^s(.)$ is parameterized by the mean vector is parameterized by the mean vector μ_i^s and the covariance matrix $\sum_i^s$ using

$$f_i^s(x) = \frac{1}{\sqrt{(2\pi)^d |\sum_i^s|}} \exp\left(-\frac{1}{2}(x - \mu_i^s)^T (\sum_i^s)^{-1}(x - \mu_i^s)\right) \quad (19)$$

Where $(\sum_i^s)^{-1}$ and $|\sum_i^s|$ denote the inverse and determinant of the covariance matrix $\sum_i^s$, respectively. The mixture weights $(\alpha_{1,}^s \alpha_{2,}^s \ldots \alpha_{M,}^s)$ satisfy the constraint $\sum_{i=1}^{M} \alpha_i^s = 1$. Collectively, the parameters of the model λ^s are denoted as $\lambda^s = \{\alpha_i^s, \mu_i^s, \sum_i^s\}$, i=1,2,…M. The number of mixture components is chosen empirically for a given data set. The parameters of GMM are estimated using the iterative expectation-maximization algorithm [19].

The motivation for using Gaussian densities as the representation of audio features is the potential of GMMs to represent an underlying set of acoustic classes by individual Gaussian components in which the spectral shape of the acoustic class is parameterized by the mean vector and the covariance matrix. Also, GMMs have the ability to form a smooth approximation to the arbitrarily shaped observation densities in the absence of other information. With GMMs, each sound is modeled as a mixture of several Gaussian clusters in the feature space. A variety of approaches to the problem of mixture decomposition have been proposed, many of which focus on maximum likelihood methods such as expectation maximization (EM) or maximum a posterior estimation (MAP). Generally these methods consider separately the question of parameter estimation and system identification, that is to say a distinction is made between the determination of the number and functional form of components within a mixture and the estimation of the corresponding parameter values.

1.) Expectation Maximization (EM)

Expectation maximization (EM) is seemingly the most popular technique used to determine the parameters of a mixture with an a priori given number of components [16]. This is a particular way of implementing maximum likelihood estimation for this problem. EM is of particular appeal for finite normal mixtures where closed-form expressions are possible such as in the following iterative algorithm by Dempster et al. (1977). The Expectation-maximization algorithm can be used to compute the parameters of a parametric mixture model distribution. It is an iterative algorithm with two steps: an expectation step and a maximization step [17]. The expectation step with initial guesses for the parameters of our mixture model, "partial membership" of each data point in each constituent distribution is computed by calculating expectation values for the membership variables of each data point [18].

That is, for each data point x_i and distribution Y_i , the membership value $y_{i,j}$ is:

$$y_{i,j} = a_i f_y(x_j; \theta_i) / f_X(x_j) \qquad (20)$$

The maximization step with expectation values in hand for group membership, plug in estimates are recomputed for the distribution parameters. The mixing coefficients a_i are the means of the membership values over the N data points.

$$a_i = 1/N \sum_{j=1}^{N} y_{i,j} \qquad (21)$$

The component model parameters θ_i are also calculated by expectation maximization using data points x_j that have been weighted using the membership values. For example, if θ is a mean μ

$$\mu_i = \sum_j y_{i,j} x_j / \sum_j y_{i,j} \qquad (22)$$

With new estimates for the $\theta_i^{'}$s, the expectation step is repeated to recompute new membership values. The entire procedure is repeated until model parameters converge.

IV. IMPLEMENTATION

A. Signal Pre-processing

Audio signal has to be pre processed 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 [8]. 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 [2].

B. Feature Extraction

Six set of features is extracted from each frame of the audio by using the feature extraction techniques. Here the low level features both time domain and frequency domain features are taken. The time domain features are ZCR and STE, the frequency domain features are spectral centroid, spectral flux, spectral roll-off and spectral entropy. An input wav file is given to the feature extraction techniques. Six set of feature values will be calculated for the given wav file. The above process is continued for 100 number of wav files. The six set of feature values for all the wav files will be stored separately for speech and music. The features are then often normalized by the computed mean value and the standard variation over a larger time unit and then stored in a feature vector [1].

C. Classification

When the feature extraction process is done the audio should be classified either as speech or music. In a more complex system more classes can be defined, such as silence or speech over music. The latter is often classed as speech in systems with only two basic classes. The extracted feature vector is used to classify whether the audio is speech or music. A mean vector is calculated for the whole audio and it is compared either to results from training data or to predefined thresholds. A method where the classification is based on the output of many frames together is proposed. In this method, based on the output the feature values are extracted from the speech/music wav file and it is appended with two categories. One category is appended for speech wav and the other category is appended for the music wav. By using the feature values with appended value SVM training is carried out. As a result of the training data two model files will be created one for speech and the other for music. The SVM trains the audio data and create two models one for speech and the other for music. For testing the feature extraction is done on different speech and music wav files other than the speech and music wav files used in the training set. All the values would be used for testing, the SVM tests the features based on models created during the training. Each second consists of 100 frames, and each frame is assigned a class by a SVM classifier. Then, a global decision is made based on the most frequently appearing class within that second.

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. A mean vector is calculated for the whole audio and is then compared either to results from training data or to predefined thresholds. A method where the classification is based on the output of many frames together is proposed. Each second consists of 100 frames, and each frame is assigned a class by a SVM classifier. The SVM will train the audio data and create two modules correspondingly. The training samples are loaded and two classes are created, for each category. The two categories will be trained with two class 0 and class 1 with 100 examples. The testing sample is tested using the trained model and create a result. The result will show whether the audio is speech or music.

Table 1: Classification Performance for different kernel function

Kernel function	*Speech*	*Music*
Gaussian	85%	88%
Sigmoidal	82%	84%
Polynomial	83%	80%

The choice of a Kernel depends on the problem at hand because it depends on what we are trying to model. The motivation behind the choice of a particular kernel can be very intuitive and straightforward depending on what kind of

information we are expecting to extract about the data. The table.1 shows that the Gaussian kernel classification performance is greater than the other two kernels.

Gaussian mixtures for the two classes are modeled for the features extracted. For classification the feature vectors are extracted and each of the feature vector is given as input to the GMM model. The distribution of the acoustic features is captured using GMM. We have chosen a mixture of 2, 4, 5, 10 mixture models. The class to which the audio sample belongs is decided based on the highest output. Audio classification using GMM gives an accuracy of 95.9%. The performance of GMM for different mixtures as shown in Fig. 6 shows that when the mixtures were increased from 5 to 10 there was no considerable increase in the performance. With GMM, the best performance was achieved with 10 Gaussian mixtures.

The performance of the system for 2, 5 and 10 Gaussian mixtures is shown in table.2. The distribution of the acoustic features is captured using GMM. The class to which the speech and music sample belongs is decided based on the highest output. Table.2 shows the performance of GMM for speech and music classification based on the number of mixtures.

Table 2: Performance of GMM for different mixtures.

GMM	*2*	*5*	*10*
Speech	92%	92%	93%
Music	88%	87%	86%

GMM and SVM systems give equivalent results for each kind of category in Table 3

Table 3: Classification results.

Category	*Speech*	*Music*
SVM	92%	95%
GMM	94%	96%

Experiments were conducted to test the performance of SVM using gaussian, sigmoidal and polynomial kernel functions. SVM performs well with a lesser number of feature vector. Using GMM, a better performance is achieved even if the size of feature vector is larger.

V. CONCLUSIONS

In this paper a system for classifying the audio into speech and music using both time domain and frequency domain is presented. SVM is trained and tested for different kernel function and performance is studied. GMM using EM algorithm is used to estimate the parameters. The performance of GMM for different mixtures shows satisfactory results.

REFERENCES

[1] Boser E. Bernhard, Guyon M. Isabelle, and Vapnik N. Vladimir. A training algorithm for optimal margin classifiers. In *5th Annual ACM Workshop on COLT*, pages 144–152. ACM Press, 1992.

[2] B. Liang, H. Yanli, L. Songyang, C. Jianyun, and W.Lingda. Feature analysis and extraction for audio automatic classification, *Proc. IEEE Int. Conf. Systems*, pages 767–772, October 2005.

[3] J. Breebaart and M. McKinney. Features for audioclassification. *Int.Conf. on MIR*, 2003.

[4] F. Gouyon, F. Pachet, and O. Delerue. Classifyingpercussive sounds: a matter of zero crossing rate. *Proceedings of the COST G-6 Conference on DigitalAudio Effects (DAFX-00)*, December 2000. Verona, Italy.

[5] Ingo Mierswa1 and Katharina Morik. Automatic feature extraction for classifying audio data,.*Machine Learning Journal*, 58(2):127–149, February 2005.

[6] C. Panagiotakis and G. Tziritas. A speech/music discriminator based on rms and zero-crossings,.*IEEE Trans. Multimedia*, 7(5):155–156, February 2005.

[7] G. Peeters. A large set of audio features for sound description. *tech. rep., IRCAM*, 2004.

[8] L. Rabiner and R.W. Schafer. Digital processing of speech signals. *Pearson Education*, 2005.

[9] Toru Taniguchi, MikioTohyama, and Katsuhiko Shirai. Detection of speech and music based on spectral tracking. *Speech Communication*, 50:547–563, April 2008.

[10] Toru Taniguchi, MikioTohyama, and Katsuhiko Shirai. Detection of speech and music based on spectral tracking. *Speech Communication*, 50:547–563, April 2008.

[11] Hongchen Jiang, Junmei Bai, Shuwu Zhang, and Bo Xu, "SVM-based audio scene classification," in Proc. IEEE Int. Conf. Natural Lang. Processing and Knowledge Engineering,,Wuhan,China,October2005,pp.131–136.

[12] J. C. Burges Christopher, "A tutorial on support vector machines for pattern recog-nition," Data mining and knowledge discovery, vol. 52, pp. 121–167, 1998.

[13] Guodong Guo and Stan Z. Li, "Content-based audio classification and retrieval by support vector machines," IEEE Trans. Neural Networks, vol. 14, no. 1, pp. 209–215, January 2003.

[14] Sourabh Ravindran, Kristopher Schlemmer, and David V. Anderson, "A physiologi-cally inspired method for audio classification," Journal on Applied Signal Processing, vol. 9, pp. 1374–1381, 2005.

[15] Menaka Rajapakse and Lonce Wyse, "Generic audio classification using a hybrid model based on GMMs and HMMs," in IEEE Int'l Conf. Multimedia Modeling, February 2005, pp. 1550–1555.

[16] Reynolds D (1993) A gaussian mixture modeling approach to text-independent speaer identification. Technical Report 967

[17] H Watanabe SM, Kikuchi H (2010) Interval calculation of em algorithmfor gmm parameter estimation. Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium pp 2686–2689

[18] Redner R, Walker H (1984) Mixture densities, maximum likelihood and the em algorithm. SIAM Review 26:195239

[19] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," SIAM Review, vol. 26, pp. 195–239, 1984.

