CHAPTER 4

AUDIO CLASSIFICATION

4.1 Introduction

Traditional methods of audio classification include distance based approaches such as K- Nearest Neighbor (KNN), rule based techniques such as using decision thresholds on probability functions. Other methods of classification are model based like Gaussian Mixture Model (GMM), and statistical learning theory based methods such as Artificial Neural Network (ANN) and Support Vector Machine (SVM). SVM classifier has gained wide popularity owing to its reduced computational complexities and greater classification accuracies. The present investigation tried all such classifiers for binary classification. The KNN based and PDF based techniques rendered reasonably acceptable results for few binary classes with SVM classifier performing the best amongst all even for multiclass situation. The researcher is therefore encouraged to continue with only SVM classifier for the work and discussion hereafter, with the target of developing a robust and reliable five class SVM based audio classifier. The five audio classes considered in the design are as shown in Fig 4.1

![Audio Classes](image)

Fig 4.1 Audio Classes
4.2 Support Vector Machine (SVM)

Given a feature vector choice, pattern classification primarily consists of the development or training of a system for classification of the different classes. A support vector machine is a powerful classifier [13] that has gained considerable popularity in recent years. The Support Vector Machine is a classifier, originally proposed by Vapnik [14] that finds a maximal margin separating hyper plane between two classes of data.

Basically, Support Vector Machine is a linear machine with some very nice properties. To explain how it works, it is perhaps easiest to start with the case of separable patterns that could arise in the context of pattern classification. In this context, the main idea of a support vector machine is to construct a hyper plane as the decision surface in such a way that the margin of separation of positive and negative examples is maximized. The machine achieves this desirable property by following a principled approach rooted in the statistical learning theory. More precisely the support vector machine is an approximate implementation of the method of structural risk minimization. This induction principle is based on the fact that the error rate of learning machine on test data (i.e. generalization error rate) is bounded by the sum of training error rate and term that depends on Vapnik-Chervonenkis (VC) dimension; in the case of separable pattern, a support vector machine produces a value of zero for the first term and minimizes the second term. Accordingly the support vector machine can provide a good generalization performance on pattern classification problems despite the fact that it does not incorporate problem-domain knowledge. This attribute is unique to support vector machines.
A notion that is central to the construction of the support vector learning algorithm is the inner-product kernel between “support vector” \( x_i \) and vector \( x \) drawn from the input space. The support vector consists of a small subset of the training data extracted from the algorithm. Depending on how this inner product kernel is generated it is possible to construct different learning machines characterized by the non linear decision surfaces of their own.

Support vector machines use supervised learning methods for classification. SVMs map input vectors to a higher dimensional space if the data is not linearly separable. Then a hyper plane is constructed to separate the input vectors. Two parallel hyper planes are constructed on each side of the hyper plane. The hyper plane that maximizes the distance between the two parallel hyper planes is found to be the solution. In linear non-separable cases, we need a kernel function to transform the original feature space to a higher dimensional space in an implicit way such that the mapped data is linearly separable. Common kernels include Polynomial, Gaussian Radial Basis Function, Sigmoid, etc. The choice of kernel is an important issue in SVM classification.

### 4.3 Optimal Hyper Plane for Linearly Separable Patterns:

Consider the training sample \{ \( x_i \), \( d_i \)\}_{i=1}^{N} where \( x_i \) is the input pattern for the \( i^{th} \) example and \( d_i \) is the corresponding desired response (target output). Let us assume that the pattern (class) represented by the subset \( d_i = +1 \) and the subset \( d_i = -1 \) are linearly separable. The equation of a decision surface in the form of a hyper plane that does the separation is

\[
w^T x + b = 0 \tag{4.3.1}
\]
where \( \mathbf{x} \) is the input vector, \( \mathbf{w} \) is an adjustable weight vector and \( b \) is the bias.

Thus,

\[
\mathbf{w}^T \mathbf{x}_i + b \geq 0 \quad \text{for } d_i = +1 \quad (4.3.2)
\]

\[
\mathbf{w}^T \mathbf{x}_i + b < 0 \quad \text{for } d_i = -1 \quad (4.3.3)
\]

The assumption of linearly separable patterns is made here to explain the basic idea behind the support vector machine. For a given weight vector \( \mathbf{w} \) and the bias \( b \), separation between the hyper plane defined in the above equation and the closest data point is called as the margin of separation. The goal of support vector machine is to find the hyper plane for which the margin of separation is maximized. Under this condition the decision surface is referred to as optimal hyper plane, as shown in Fig 4.2.

Fig 4.2 Geometric construction of optimum hyper plane for two dimensional input space
Let \( \mathbf{w}_0 \) and \( b_0 \) be the optimum values of weight vectors and bias respectively.

Correspondingly, the optimum hyper plane representing multidimensional linear decision surface in input space, is defined by,

\[
\mathbf{w}_0^T \mathbf{x} + b_0 = 0
\]  

(4.3.4)

The discriminant function

\[
g(\mathbf{x}) = \mathbf{w}_0^T \mathbf{x} + b_0
\]

(4.3.5)

gives an algebraic measure of the \textit{distance} from \( \mathbf{x} \) to optimum hyper plane [50]. The easiest way to see this is to express \( \mathbf{x} \) as

\[
\mathbf{x} = \mathbf{x}_p + r \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|}
\]

(4.3.6)

Where \( \mathbf{x}_p \) is the normal projection of \( \mathbf{x} \) onto the optimum hyper plane, \( r \) is the desired algebraic distance; \( r \) is positive if \( \mathbf{x} \) is on the positive side of hyper plane and negative if \( \mathbf{x} \) is on the negative side. Since, by the definition \( g(\mathbf{x}_p) = 0 \), it follows that

\[
g(\mathbf{x}) = \mathbf{w}_0^T \mathbf{x} + b_0 = r \|\mathbf{w}_0\|
\]

(4.3.7)

Or

\[
r = \frac{g(\mathbf{x})}{\|\mathbf{w}_0\|}
\]

(4.3.8)

In particular the distance from origin (i.e. \( \mathbf{x} = 0 \)) to the hyper plane is given by \( \frac{b_0}{\|\mathbf{w}_0\|} \).

If \( b_0 > 0 \), origin is on the positive side of the hyper plane; if \( b_0 < 0 \) then it is on negative side. If \( b_0 = 0 \) optimal hyper plane passes through origin. Geometric interpretation of these algebraic results is shown in Fig 4.3.
The issue is to find parameters $\mathbf{w}_0$ and $b_0$ for the optimum hyper plane given the training set $\{ \mathbf{x}_i, \ d_i \}_{i=1}^N$. It is clear that the pair $(\mathbf{w}_0, b_0)$ must satisfy the constraints:

\[
\begin{align*}
\mathbf{w}_0^T \mathbf{x} + b_0 &\geq 1 & \text{for } d_i = +1 \quad (4.3.9) \\
\mathbf{w}_0^T \mathbf{x} + b_0 &\leq -1 & \text{for } d_i = -1 \quad (4.3.10)
\end{align*}
\]

The particular data points $(\mathbf{x}_i, \ d_i)$ for which the first or the second line of the equation (4.3.9) and (4.3.10) is satisfied with an equality sign are called as support vectors, hence the name support vector machine. In conceptual terms, support vectors are those data points that lie closest to the decision surface and are most difficult to classify. Thus they have direct bearing on optimum location of the decision surface.

Consider a support vector $\mathbf{x}^{(s)}$ for which $d^{(s)}=+1$. Then by definition,

\[
g(\mathbf{x}^{(s)}) = \mathbf{w}_0^T \mathbf{x}^{(s)} \pm b_0 \quad \text{for } d^{(s)} = +1 \quad (4.3.11)
\]

Equation (4.3.8) the algebraic distance from support vectors $\mathbf{x}^{(s)}$ to the optimal hyper plane is
\[
\begin{align*}
    r &= \frac{g(x^{(s)})}{\|w_0\|} = \begin{cases} 
    \frac{1}{\|w_0\|} & \text{if } d^{(s)} = +1 \\
    -\frac{1}{\|w_0\|} & \text{if } d^{(s)} = -1 
\end{cases}
\end{align*}
\] (4.3.12)

Where, the plus sign indicates that \(x^{(s)}\) lies on the positive side of the hyper plane, negative sign indicate that \(x^{(s)}\) lies on the negative side of the hyper plane. Let \(\rho\) denotes the optimum value of the margin of separation between the two classes that constitute the training set. Then from the equation (4.3.12) it follows that

\[
\rho = 2r = \frac{2}{\|w_0\|} \quad (4.3.13)
\]

This states that maximizing the margin of separation is equivalent to minimizing the Euclidean norm of the weight vector. Thus optimum hyper plane defined by equation (4.3.4) is unique in the sense that the optimum weight vector \(w_0\) provides the maximum possible separation between the positive and negative examples. This optimum condition is achieved by minimizing the Euclidean norm of the weight vector \(w\).

The constrained optimization problem to be solved is as stated below.

*Given the training sample \(\{x_i, d_i\}_{i=1}^N\), find the optimum values of the weight vector \(w\) and bias \(b\) such that they satisfy the constraints*

\[
d_i (w^T x_i + b) \geq 1 \quad \text{for } i = 1, 2, \ldots, N \quad (4.3.14)
\]

*And weight vector \(w\) minimizes the cost function*

\[
\phi(w) = \frac{1}{2} w^T w \quad (4.3.15)
\]

*This constrained optimization problem is called primal problem.*
4.4 Hyper Plane for Non-Separable Patterns

The discussion thus far has focused on linearly separable patterns. The more difficult case of non separable patterns is considered here. Given such a training data, it is not possible to construct a separating hyper plane without encountering classification errors. An optimum hyper plane is to be found that minimizes the probability of classification error, averaged over a training set.

Fig 4.4 Non-separable training sets introduces misclassification

The margin of separation between classes is said to be soft if data point \((x_i, d_i)\) violates the condition given in equation (4.3.14).

To set the stage for formal treatment of non separable data points, a new set of non negative scalar variables \(\{\xi_i\}_{i=1}^N\) is introduced in the definition of separating hyper plane as given in equation (4.4.1).

\[
d_i(w^T x_i + b) \geq 1 - \xi_i \quad \text{for } i = 1, 2, \ldots, N \quad (4.4.1)
\]
The $\xi_i$ are called \textit{slack variables}, they measure the deviation of data point from the ideal condition of pattern separability.

The support vectors are those particular data points that satisfy equation (4.4.1) precisely even if $\xi_i > 0$. The primal problem in case of non separable case may thus be formally defined as follows, where $C$ is user specified positive parameter also called as SVM penalty/cost parameter.

\textit{Given the training sample } $\{x_i, d_i\}_{i=1}^N$, \textit{find the optimum values of the weight vector } $w$ \textit{and bias b such that they satisfy the constraints}

$d_i(w^T x_i + b) \geq 1 - \xi_i$ \hspace{1cm} \text{for } i = 1, 2, ..., N \hspace{1cm} (4.4.3)

\textit{And weight vector } $w$ \textit{minimizes the cost function}

$\varphi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i$ \hspace{1cm} (4.4.4)

The parameter $C$ controls the tradeoff between complexity of the machine and the number of the non separable points; it may be therefore viewed as regularization parameter. The parameter $C$ has to be selected by the user. This can be done in one of the two ways. The parameter $C$ is determined experimentally via the standard use of training / (Validation) test set, which is crude form of re-sampling or it can be determined analytically.
4.5 Building a SVM for Pattern Recognition

For patterns that are not linearly separable, the following mathematical operations are performed in construction of SVM optimal hyper plane.

1. Non linear mapping of input vector into high-dimensional feature space that is hidden from both input and output. The low dimensional input data $\mathbf{x}$ is mapped into a high-dimensional feature space by mapping function $\varphi(\mathbf{x})$.

2. Construction of optimum hyper plane for features obtained for separating the features discovered in step 1.

These two mathematical operations are illustrated in Fig 4.5.

![Non linear mapping from the input space to higher dimension feature space](image)

*Fig 4.5 Non linear mapping from the input space to higher dimension feature space*

The separating hyper plane is now defined as a linear function of vectors drawn from the feature space rather than original input space. Consider an input space made up of non-linearly separable patterns. Cover’s theorem states that such a multidimensional space may be transformed into new feature space where the patterns are linearly
separable with high probability provided two conditions are satisfied. First the
transformation is nonlinear. Second, the dimensionality of the feature space is high
enough. These two conditions are embodied in operation 1. However, Cover’s theorem
does not discuss the optimality of separating hyper plane.

The separating hyper plane is now defined as a linear function of vectors drawn
from the feature space rather than original input space. Most importantly, construction of
this hyper plane is performed in accordance with the principal of structural risk
minimization that is rooted in VC dimension theory. The construction hinges on the
evaluation of inner product kernel.

**Inner product kernel:** The term \( \phi^T(x_i)\phi(x) \) represents the inner product of two vectors
induced in the feature space by the input vector \( x \) and the input pattern \( x_i \) pertaining to
the \( i \)-th example the inner product kernel denoted by \( K(x, x_i) \) and defined by

\[
K(x, x_i) = \phi^T(x)\phi(x_i) \quad \text{for } i = 1, 2, \ldots, N \quad (4.5.1)
\]

The inner product kernel is a symmetric function of its arguments as shown by

\[
K(x, x_i) = K(x_i, x) \quad (4.5.2)
\]

The requirement on kernel function is to satisfy Mercer’s theorem. In table 4.1 the inner
product kernels for common types of support vector machines are summarized.
### Table 4.1: Summary of Inner Product Kernels in SVM

<table>
<thead>
<tr>
<th>Type of SVM</th>
<th>Inner Product Kernel</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$\mathbf{x}^T \mathbf{x}_i$</td>
<td>--</td>
</tr>
<tr>
<td>Polynomial</td>
<td>$(\gamma \mathbf{x}^T \mathbf{x}_i + 1)^p$</td>
<td>Degree $p$ and $\gamma$ are kernel parameters defined by the user ($\gamma &gt; 0$)</td>
</tr>
<tr>
<td>Radial Basis Function</td>
<td>$e^{-\gamma |\mathbf{x} - \mathbf{x}_i|^2}$</td>
<td>The width $\gamma$ is kernel parameter specified a priori by the user ($\gamma &gt; 0$)</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>$\tanh(\gamma \mathbf{x}^T \mathbf{x}_i + 1)$</td>
<td>--</td>
</tr>
</tbody>
</table>

#### Architecture of Support Vector machine

In the conventional approach, model complexity is controlled by keeping the number of features small. On the other hand, the support vector machine offers a solution to the design of a learning machine by controlling model complexity independently of dimensionality.

- **Conceptual problem:** Dimensionality of the feature space is purposely made very large to enable construction of decision surface in the form of hyper plane. The model complexity is controlled by imposing certain constraints on the construction of separating hyper plane, which results in extraction of fraction of training data as support vectors.

- **Computational problem:** Numerical optimization in high dimensional feature space suffers from the curse of dimensionality. This can be avoided by using inner-product kernel and solving the dual form of the problem.
Thus, support vector machine has the inherent ability to solve a pattern classification problem in a manner close to the optimum for the problem of interest. Moreover, it is able to achieve such a remarkable performance with no problem domain knowledge built into the design of the machine. Fig 4.6 shows the architecture of a support vector machine.

![Fig 4.6 SVM Architecture](image)

**4.6 LIBSVM**

In this work, **LIBSVM 2-89** software is used [18]. It is efficient software for SVM classification and regression. It can handle different types of SVM, namely, C-SVM classification, nu-SVM classification, one-class-SVM, epsilon-SVM regression, and nu-SVM regression.
The following steps are used in order to get a good accuracy:

- Data Processing.
- Scaling.
- Model Selection.
- Cross-validation and Grid-search
- Train classifier using the best values of $C$ and $\gamma$.
- Test/predict.

These points are explained in detail further.

**Data Processing:**

SVM requires that each data instance is represented as a vector of real numbers. Hence, if there are categorical attributes, they are first converted into numeric data. Numbers are used to represent an m-category attribute. See the Appendix-I for a typical training file format.

**Scaling:**

Scaling data before applying SVM is very important. The main advantage is to avoid features/attributes in greater numeric ranges dominate those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation. Because kernel values usually depend on the inner products of feature vectors, e.g. for the linear kernel and the polynomial kernel, large attribute values might cause numerical problems. Linearly scaling each attribute to the range $[-1:1]$ or $[0:1]$ is supported and recommended by the software. It is used for scaling the training and test data. Scaling of values helps to have values spread over a long range, compressed within the desired values. This helps in the improvement of accuracy.
Model Selection:

Though there are only four common kernels mentioned in Table 4.1, one must decide which one to try first. Then the penalty parameter $C$ and kernel parameters are chosen. It is observed that in general RBF is a reasonable first choice. The RBF kernel nonlinearly maps samples into a higher dimensional space, so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. Furthermore, the linear kernel is considered as a special case of RBF [13]. In addition, the sigmoid kernel behaves like RBF for certain parameters [62].

The second reason is the number of hyper parameters which influences the complexity of model selection. The polynomial kernel has more hyper parameters than the RBF kernel. Finally, the RBF kernel has less numerical difficulties. However, there are some situations where the RBF kernel is not suitable. In particular, when the number of features is very large, one may just use the linear kernel.

Cross-validation and Grid-search:

There are two parameters while using RBF kernels: $C$ and $\gamma$. It is not known beforehand which $C$ and $\gamma$ values are the best for one problem; consequently some kind of model selection (parameter search) must be done. The goal is to identify good $(C, \gamma)$ so that the classifier can accurately predict unknown data (i.e., testing data). Note that it may not be useful to achieve high training accuracy (i.e., classifiers accurately predict training data whose class labels are indeed known). Therefore, a common way is to separate training data into two parts of which one is considered unknown in training the classifier. Then the prediction accuracy on this set can more precisely reflect the
performance on classifying unknown data. An improved version of this procedure is cross-validation.

In v-fold cross-validation, the training set is first divided into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining v-1 subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified. The cross-validation procedure can prevent the over fitting problem. It is recommended to use “grid-search” on C and γ using cross-validation.

“Grid.py” is a program which performs a “grid-search” on C and γ using cross-validation. Basically pairs of (C, γ) are tried and the one with the best cross-validation accuracy is picked. An exponentially growing sequences of C and γ is a practical method to identify good parameters (for example, C = 2^-5, 2^-3, …, 2^15; and γ = 2^-15, 2^-13, …, 2^3). The grid-search is a straightforward approach to determine the optimum values of C and γ.

There are three reasons of preferring grid-search approach over other methods:

1. It does an exhaustive parameter search by approximations or heuristics.
2. The computational time to find good parameters by grid-search is comparable to that by advanced methods, since there are only two parameters to be determined.
3. Unlike the advanced iterative processes, grid-search can be easily parallelized because each (C, γ) is independent.

A graphical plot is generated that depicts the variation of log\(_2\)C versus log\(_2\)γ. A typical graphical plot is shown in Fig 4.7.
The classification results using SVM classifier are shown in next section. Some very important SVM commands used in LIBSVM are listed in Appendix-II.

4.7 Results of Audio Classification

4.7.1 Database

The Speech and Music files in this database are provided by Prof. Dan Ellis of Columbia University. Database consists of 150 music files, 150 speech files each of male speech from the different age male speakers and different age female speakers. The data of noise is downloaded from the noise database available on internet and silence files are recorded using PRAAT software [65]. All files in database have sampling frequency of 16 KHz, 16 bit mono and stored in .wav file format. The 90 files of each class (Total 450=
90*5 files) are randomly selected for training and remaining 60 files of each class (Total 300=60*5 files) are used for testing.

4.7.2 Feature Selection Based on their Performance in Binary Classification

Algorithms are developed in MATLAB platform to compute the five frame level features, Zero Crossing Rate (ZCR), Short Time Energy (STE), Short Time Fourier Transform (STFT), Spectral Centroid (SC), Pitch Frequency (PF) for frame size of 16-20 ms.

The seven clip level features such as mean ZCR, High Zero Crossing Rate Ratio (HZCRR), mean STE, Low Short Time Energy Ratio (LSTER), Spectrum Flux(SF) , Standard Deviation of Spectral Centroid (SDSC) and Pitch Degree (PD) are then computed based on frame level features as discussed in section 3.3. The duration of clip is 1 second. The clip is considered as classification unit.

The suitability of a particular feature for a particular binary classification is initially tested using conventional threshold based/Rule based and K nearest neighbor classification methods. The binary class combination is selected from five audio classes under consideration. The feature(s) whose performance is competent enough to get higher classification accuracy in two class classification are explained with the help of Probability Density Function (PDF) plots which are fitted in normal distribution by using ‘dfittool’ of MATLAB.
1. Features for Music-Speech discrimination

A. Standard Deviation of Spectral Centroid

The Fig 4.8 shows the probability density function of standard deviation of spectral centroid for the two classes, speech and music. The spectral contents of speech varies more drastically from frame to frame. Music, on the other hand, has fewer variations in spectral contents, i.e. once a particular musical note is attained, the spectral contents remains same throughout with minor variations.

B. Spectrum Flux

Fig 4.9 PDF plot of SF of Music and Speech
(Mean value of SF of Music 1.32, mean value of SF of speech 1.58)
The Fig 4.9 shows the probability density function of spectrum flux for the two classes, speech and music. The spectral contents of speech varies more drastically from frame to frame.

C. High Zero Crossing Rate Ratio

![Figure 4.10 PDF plot of HZCRR of Music and Speech](image)

(Mean value of HZCRR of Music 0.118, mean value of HZCRR of speech 0.189)

The Fig 4.10 shows the probability density function of HZCRR for the two classes, speech and music. As music is more systematic than a speech the frame having ZCR more than mean ZCR value of the clip are less. So HZCRR of music is less than that of speech.

2. Features for Speech-Silence discrimination

A. Spectrum Flux

![Figure 4.11 PDF Plot of SF of silence and speech](image)

(Mean value of SF of Silence 0.88, mean value of SF of speech 1.58)
The Fig 4.11 shows the probability density function of spectral flux for the two classes, speech and silence. The spectral contents of speech varies more drastically from frame to frame. Silence, on the other hand, has very few spectral contents, thus having a lower value of spectral flux.

B. Short Time Energy for the silence clips is very close to zero and hence it is always one of the best features to discriminate between silence and speech.

3. Speech-Noise discrimination

A. Zero Crossing Rate

![PDF Plot of ZCR of speech and noise](Image)

(Mean value of ZCR of Speech 0.13, mean value of ZCR of Noise 0.43)

The Fig 4.12 shows the probability density function of zero crossing rate for the two classes, speech and noise. Noise signal will have more zero crossings (i.e. sign changes) as compared to the speech signal. Thus, the noise signal has a higher value of ZCR.

4. Music Silence Discrimination:

A. Short Time Energy for the silence clips is very close to zero and hence it is always one of the best features to discriminate between silence and music
B. Standard Deviation of Spectral Centroid: As expected the mean value of SDSC for music is lower than that of noise.

6. Silence-Noise discrimination

A. Zero Crossing Rate

![Fig 4.13 PDF Plot of ZCR of Silence and Noise](mean value of ZCR of Silence 0.08, mean value of ZCR of Noise 0.43)

The Fig 4.13 shows the probability density function of zero crossing rate for the two classes, silence and noise. Noise signal will have more zero crossings as compared to the silence signal. Thus, the noise signal has a higher value of ZCR.

5. Music- Noise discrimination

A. Low Short Time Energy Ratio

![Fig 4.14 PDF Plot of LSTER of Music and Noise](Mean value of LSTER for Noise 0.091, mean value LSTER 0.39)
The Fig 4.14 shows the probability density function of LSTER for the two classes, music and noise. It is observed that mean LSTER value for noise is less than music.

B. Zero Crossing Rate: It is also observed that ZCR of music is less than noise.

![PDF Plot of ZCR of Music and Noise](image)

(Mean value of ZCR for Music 0.084, mean value of ZCR for Noise 0.43)

The Fig 4.15 shows the probability density function of zero crossing rate for the two classes, music and noise. Noise signal has more zero crossings as compared to the music signal.

7. Male speech-Female speech discrimination

A. Pitch Degree

![PDF Plot of Pitch Degree of Male Speech and Female Speech](image)

(Mean value of Male PD 134.6Hz, mean value of Female PD 176.5 Hz)
The Fig 4.16 shows the probability density function of pitch degree for the two classes, male speech and female speech. Female voice has a higher pitch frequency as compared to the male voice. Thus, the pitch degree value is higher for females than for males. To improve the classification accuracy between these two classes, features like HZCRR and LSTER are clubbed with pitch in SVM experimentation.

4.7.3 SVM Based Two Class Classification

Algorithm for SVM based classification: The steps involved in development of SVM based binary classification are explained by taking example of speech music classification using SDSC and SF as the features.

1) Preparation of training file: Out of 150 files of each class randomly select 90 files for training. Prepare a training file train.txt using best ranked features from the results in section 4.7.2. In case of speech-music classifier SDSC and SF are selected as the clip level features. A training file thus consist features of total 180 training samples (90 of each class). A typical training file for above mentioned classes and features is given in Appendix-I.

2) Scaling of Data: Conduct a simple scaling on data which normalizes the variation of feature values in a range of [0:1]. So that effect of all the features will be equally considered in the design of classifier.

3) Estimate optimum value of C and \( \gamma \) by using grid.py search program. Optimum value of C and \( \gamma \) is essential in order to have minimum prediction error at moderate Model complexity.

4) Use optimum values of C and \( \gamma \) for a Kernel function and train a SVM Classifier using the train.scale file as an input. Use 10-fold cross-validation.
5) **Obtain SVM model.out file** for the classifier. The contents of model file are SVM type, Kernel type, number of classes and number of support vectors.

6) **Testing:** Perform operations 1 and 2 for preparing *test.scale* file which includes 120 test samples (60 of each class). Use the SVM predict program to find the correct classification accuracy.

The results showing files used and classification accuracy achieved for various two class combination using LIBSVM interfaced with MATLAB GUI is shown in Fig 4.17. Total numbers of correctly classified samples are 118 out of 120. Classification accuracy is 98.33%.

![Fig 4.17 Results of Speech –Music Classification using SDSC and SF as features (Classification accuracy =98.33%, Kernel Type RBF, C=32, γ =2)](image)

In a similar manner the binary classification results using SVM classifier are evaluated and are given in Fig 4.18 to 4.23.
Fig 4.18 Results of Speech – Silence Classification using STE and SF as features
(Classification accuracy =98.33%, Kernel Type RBF/Linear, C=512, γ =0.5)

Fig 4.19 Results of Speech – Noise Classification using ZCR as a feature
(Classification accuracy =100%, Kernel Type Linear, C=0.03125, Gamma =0.0078125)
Fig 4.20 Results of Music Silence Classification using STE as a feature (Classification accuracy =98.33%, Kernel Type Linear/RBF, C=1024, Gamma =0.5)

Fig 4.21 Results of Music Noise Classification using SDSC and LSTER as features (Classification accuracy =100%, Kernel Type Linear, C=32, Gamma =0.007)
Fig 4.22 Results of Silence Noise Classification using ZCR as a feature
(Classification Accuracy =100%, Kernel type RBF, C=0.031, gamma =0.007)

Fig 4.23 Results of Male Female Speech Classification using PD, HZCRR, LSTERS as features
(Classification Accuracy =93.33%, Kernel type RBF, C=128, gamma =8)
Table 4.2: Two class classification results using SVM

<table>
<thead>
<tr>
<th>Classes</th>
<th>Features</th>
<th>Kernel type</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech/Music</td>
<td>SDSC + SF</td>
<td>RBF</td>
<td>98.33 %</td>
</tr>
<tr>
<td>Speech/Silence</td>
<td>SF + STE</td>
<td>Linear/RBF</td>
<td>98.33 %</td>
</tr>
<tr>
<td>Speech/Noise</td>
<td>ZCR</td>
<td>Linear</td>
<td>100 %</td>
</tr>
<tr>
<td>Music/Silence</td>
<td>STE</td>
<td>Linear/RBF</td>
<td>98.33 %</td>
</tr>
<tr>
<td>Music/Noise</td>
<td>SDSC + LSTER</td>
<td>Linear</td>
<td>100 %</td>
</tr>
<tr>
<td>Silence/Noise</td>
<td>ZCR</td>
<td>RBF</td>
<td>100 %</td>
</tr>
<tr>
<td>Speech Male/female</td>
<td>PD + HZCRR + LSTER</td>
<td>RBF</td>
<td>93.33 %</td>
</tr>
</tbody>
</table>

4.7.4 Multiclass Classification Results

There are two popular approaches of multiclass classification using LIBSVM.

(1) **one against many** where each category is split out and all of the other categories/classes are merged.

(2) **one against one** where k(k-1)/2 models are constructed where k is the number of categories/classes. LIBSVM uses this approach owing to the reduced training time.

The Fig 4.24 shows the result of five class classification with a classification accuracy of 96.33% using RBF kernel and seven clip level features (289 out of 300 test samples are predicted correctly by the SVM classifier). A GUI developed for binary and multiclass SVM classification is also shown.
Fig 4.24 Results of Multi Class Classification using best ranked features (Classification Accuracy = 96.33%, Kernel type RBF, C=2, gamma =2)

Table 4.3: Multiclass Classification results using SVM

<table>
<thead>
<tr>
<th>Features used</th>
<th>Kernel type</th>
<th>C parameter</th>
<th>Gamma</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCR + HZCRR +</td>
<td>Linear</td>
<td>2</td>
<td>2</td>
<td>93.33 %</td>
</tr>
<tr>
<td>STE + LSTER +</td>
<td>Polynomial</td>
<td>2</td>
<td>2</td>
<td>93.33 %</td>
</tr>
<tr>
<td>SDSC + SF + PD</td>
<td>RBF</td>
<td>2</td>
<td>2</td>
<td>96.33 %</td>
</tr>
</tbody>
</table>
4.8 Results of Audio Segmentation

For audio segmentation, based on feature vector extraction theory discussed in (section 3.4), an algorithm for computing Kullback-Leibler Divergence Distance (KLD) measure is developed.

Algorithm for KLD computation:

- Get the speech signal and its sampling frequency $F_s$.
- Set frame length as 20 ms and window length as 3 seconds and overlap of 1 second.
- Divide the input speech into frames and windows.
- Obtain $10^{th}$ order Linear Predictive Coefficients for each frame.
- Compute the LSP values from Linear Predictive Coefficients.
- Obtain mean and covariance matrix ($C$) of LSP’s, from LSP values all frames in a window.
- Obtain difference of covariance matrix of LSP’s, for two windows $i$ and $j$ ($C_i - C_j$). ($1 < i, j < \text{Number of windows}$)
- Obtain difference of inverse of covariance matrix of LSP’s, for clips $i$ and $j$ ($C_i^{-1} - C_j^{-1}$).
- Find KL distance $D_{ij}$ between two clips as
  $$D_{ij} = tr\left[ (C_i - C_j)(C_j^{-1} - C_i^{-1}) \right] \quad (4.8.1)$$
- Select threshold value $D_{threshold}$ equal to mean of the values of matrix $D_{ij}$.
  $D_{ij}$ matrix is symmetric and the values of all diagonal elements are zero.
- Apply thresholding,
  
  if $D_{ij} < D_{threshold}$; set KLD=0 assign that element of $D_{ij}$ a black color.
  
  else if $D_{ij} \geq D_{threshold}$; set KLD=1 assign that element of $D_{ij}$ a white color.
- Display $D_{ij}$ as binary image (0, 1), as it is easy to segment the consecutive clips of black colors representing one class.

Fig 4.25(a) shows audio signal of 60 seconds: Speaker1 and Speaker2 30 seconds each.

Fig 4.25(b) shows the binary image of matrix $D_{ij}$ (KL divergence distance) obtained by
applying KLD algorithm to the audio signal in (4.25(a)). It can be clearly seen that figure is symmetric about diagonal and there are two segments in this audio clip. The separating point is at window number 30 corresponding to speaker change at 30 seconds.

![Figure 4.25](image1.png)

(a) Audio signal of 60 seconds: Speaker$_1$ and Speaker$_2$ 30 seconds each
(b) Segmentation based on $D_{ij}$ for Speaker$_1$-Speaker$_2$ (0-30, 31-60: two segments)

Similarly the results of interlaced speech-music clips are shown in Fig 4.26 to 4.30.

![Figure 4.26](image2.png)

(a) Audio signal of 40 seconds: Speech 30 seconds and Music 10 seconds
(b) Segmentation based on $D_{ij}$ for Speech-Music (0-30, 31-40: two segments)
Fig 4.27 (a) Audio signal of 60 seconds: Speech, Music and Speech 20 seconds each
(b) Segmentation based on $D_{ij}$ for Speech-Music-Speech (three segments)

Fig 4.28 (a) Audio signal of 60 seconds: Music, Speech and Music 20 seconds each
(b) Segmentation based on $D_{ij}$ Music-Speech-Music (three segments)
Fig 4.29 (a) Audio signal of 60 seconds: Music, Speech, Music and Speech 15 seconds each (b) Segmentation based on $D_{ij}$ Music-Speech-Music-Speech (four segments)

Fig 4.30 (a) Audio signal of 60 seconds: Speech, Music, Speech and Music 15 seconds each (b) Segmentation based on $D_{ij}$ Speech-Music-Speech-Music (four segments)
4.9 Results of Voiced/Unvoiced Discrimination

Speech can be basically classified into two main categories, voiced and unvoiced. Voiced speech is produced essentially by the vibrations of the vocal cord. They are associated with a fundamental frequency, $F_o$, which denotes the periodic opening and closing of vocal cords. Unvoiced speech, on the other hand, is not associated with vocal cord vibrations and generally belongs to the following two categories: fricatives and plosives (as discussed in section 3.4.1).

A comparative performance study of five voiced/unvoiced classification algorithms has been conducted during the research. The algorithms in this study are 1) Pitch detection by autocorrelation method 2) Discrete Wavelet Transform (DWT) method 3) Linear Predictive Coding (LPC) method 4) Log-energy method 5) ZCR method.

The best results for V/UV frame classification was obtained by using pitch detection method and discrete wavelet transform based sub-band energy ratio method. The results of these two methods are combined (using AND operation for each frame) for double check of correct V/UV discrimination.

Table 4.4 gives the accuracy in percentage for correct V/UV classification using “AND operation” between Pitch Detection algorithm AND DWT based algorithm. The wavelets used for DWT based algorithm are Harr, Bi-orthogonal 6.8 and Coiflet 5 wavelets. The performance was initially tested with clean speech and later on a noise (in 0-500 Hz band) of 5dB and 15dB was added to the clean speech.
Table 4.4: Results of Voiced /Unvoiced classification accuracy

<table>
<thead>
<tr>
<th>Speech Signal</th>
<th>Pitch AND DWT (Haar wavelet)</th>
<th>Pitch AND DWT (Bior 6.8 wavelet)</th>
<th>Pitch AND DWT (Coiflet 5 wavelet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean speech</td>
<td>95%</td>
<td>91%</td>
<td>91%</td>
</tr>
<tr>
<td>Clean speech + 5dB Noise</td>
<td>90%</td>
<td>94%</td>
<td>93%</td>
</tr>
<tr>
<td>Clean speech + 15dB Noise</td>
<td>93%</td>
<td>90%</td>
<td>87%</td>
</tr>
</tbody>
</table>

4.10 Summary:

The conclusions drawn on the basis of the classification and segmentation methods implemented in this chapter are:

The features having better interclass discriminatory power are initially identified using rule based method. The performance of the same features was tested using SVM classifier. The performance of two class SVM classification was better compared to rule based and KNN based techniques. SVM classifiers with RBF kernels are recommended for most of the cases because they can solve non-linear problems and they have lower model complexity as compared to polynomial kernels. The grid search program and tenfold cross validation helps to find optimum values of hyper parameters of the selected kernel in order to reduce prediction error at moderate complexity.

A multiclass classification using best ranked features and optimum hyper parameters for a RBF kernel based SVM classifier gives 96.33% accuracy for aforementioned five classes. Speech signal was further classified into V/UV using DWT and pitch detection method which gives accuracy 95%. KLD distance is the best measure to detect potential speaker change in multi-speaker audio recording. This measure has also given satisfactory results for segmenting interlaced speech- music clips.