Chapter 5

Related Pattern Recognition basics

5.1 Introduction

Human beings are the best pattern recognizers. They have a skill of recognizing and identifying a person from many people even after so many years of gap with different light conditions, different aging, viewing conditions with varying face and voice expressions. This excited many researchers in this field to focus on the pattern recognition tasks to develop and make the computer system as an intelligent as the human being.

A set of observations and measurements of an object is called a feature. Feature vector is the combination of such type of n features of the object framed into n dimensional column vector. A group of objects which comes under one category is called class. Collection of features with correct class information of an object is called a pattern. For instance face image, speech signal, bar code, fingerprint image, a word etc come under examples of patterns. Observing and extracting multiple features is a complex task for machines and less complex task(in some cases it is obvious) for humans [58].

The ‘Pattern recognition’ is a study of Observing and extracting the patterns by machines and is used to classify the input pattern into a particular class. Basically this system consists of two stages. First stage is analysis which is used to extract the feature from the pattern and the second stage is used for classification.
This stage recognizes, the objects based on the features. For classification stage, the training set is needed. The training strategies consists of supervised learning and unsupervised learning \[58\] \[86\]. For supervised learning prior class information is needed. No prior class information is needed for unsupervised learning.

5.2 Categorization of classification models

The pattern recognition systems are mainly classified into four major methodologies.

- Statistical approach
- Syntactic or structural approach
- Template matching
- Neural networks

5.2.1 Statistical approach

Statistical approach mainly depends on statistical properties and probabilities. Patterns are represented as a group of features. Each feature is converted in the form of numbers and placed into a vector \[47\]. A pattern has many such features. Each feature set is represented as a point in a multidimensional space. For instance, if we have \(n\) features it is represented as a point in \(n\) dimensional vector space. The statistical approach measures the distance between the points in the multidimensional space through which it would classify the patterns. This approach establishes a boundary to separate various class patterns. The effectiveness depends on ‘how best the various class patterns are separated’.
5.2.2 Syntactic or Structural approach

The syntactic approach depends on the complex relation between the features in a pattern. In this structural approach, the pattern is represented as a hierarchical structure composed of substructures. All patterns related to a single class contain the same structural properties, which means that the complex pattern is divided into small sub-patterns and each smallest sub-pattern is called a primitive (also called codeword). So in this approach, shape is represented as a set of predefined primitives which is called a code book.

5.2.3 Template matching

Template matching is one of the simplest and widely used approach in pattern recognition to recognize shapes in the image. In template matching, we should know the template of the pattern to be identified. This template is obtained from the training set of samples. The test pattern to be identified is the match of the saved template with each possible position, each possible rotation or each other geometric transformation of the template and has to compare each neighboring pixels with this template. This approach calculates the similarity between the two points, curves and shapes of the same type.

5.2.4 Neural Networks

Neural Network method is parallel to computing system consisting of a structure with large number of processors and interconnections between them. The interconnection sends the signal between one processor to the other. The Neural Network trained itself and solves the complex problems depending upon the available knowledge. Neural networks inspired by the physiological information of the human brain and its structure resembles like a brain nervous system.
5.3 Selected classification techniques

The emotion recognition requires a database with a significant number of variables, which means a high dimensionality database is needed. This high dimensional database contains more similar features. In such cases, we need to reduce the dimensionality by selecting the non-correlated features (information loss is less) from the database. If we select inadequate features, the accuracy would be diminished. So we need to acquire the complete knowledge about similarities and differences in the given data. The statistics are based on analyzing the high dimensional data in terms of relationship between the variables in the data set. Each technique contains train and test databases. The train database contains various speech samples of different emotions. Test database contains one speech sample of an unknown emotion. Each technique computes the basis vector by using some statistical properties. The Statistical approach based pattern recognition models are discussed in this thesis. This approach is simple and most widely used in practice for high dimensional data. The structural approach is very complicated for high dimensional data. The template matching methods requires high processing time to calculate the template. In case of Neural Networks, the conventional concepts (number of layers, number of nodes, learning rate etc.) are not robust enough and depend on the complexity of the task. We cannot easily formulate a mathematical definition of the problem to be solved.

The statistical techniques are extensively used in various applications like Signal Processing and Image Processing. The Linear Discriminant Analysis(LDA) and Regularized Discriminant Analysis(RDA) use the covariance or correlation matrix of given data samples. The LDA and RDA are used for multidimensional data besides being used in the dimensionality reduction (without any information loss) methods.

The basis vector is very high in dimension, which needs to be reduced. After forming the basis vector, the feature vector is computed by projecting the
train database speech samples into the basis vector. Then matching is done by using the distance measurements. In this work, two sets of data are taken. The first one, is for training and the second one, is for testing. The training set contains data samples in folders depending upon the selection of the data base. Each folder contains emotional speech samples of one class. Test set contains a group of speech samples used for testing. For this, the test speech samples are chosen from various emotion states. All this process is carried out using MATLAB. The statistical approach performs well for various pattern recognition tasks in both supervised and unsupervised types. The selected pattern recognition techniques discussed in this thesis are

- Linear Discriminant Analysis
- Regularized Discriminant Analysis
- Support Vector Machine
- k Nearest Neighbor

### 5.3.1 Linear Discriminant Analysis (LDA)

The classical Linear Discriminant Analysis (LDA) is also called Fisheries Linear Discriminant (FLD). This method was developed in 1936 by Robert Fisher. In this method, training and test samples were projected into the same subspace and similarities between these data samples were identified. The LDA is ‘appearance based’ linear subspace technique. So it uses statistics like mean and covariance. The computation of the mean and co-variance is performed by using the train data samples to form the data matrix X. The computations of mean and co-variance are as shown in Eq. 5.2 and Eq. 5.3. The Fisheries Linear Discriminant Analysis technique is explained in the Fig. 5.1.
5.3.1.1 Computation of mean and covariance matrix

Initially we consider the speech sample in the training data set and convert this sample into a column vector. On doing so, this will convert all the speech samples in the training set into column vectors. For instance if the training set contains $N$ speech samples and size of each speech sample is $M \times 1$, after converting all the speech samples into column vectors and then appending all the columns forms a two dimensional data matrix $X$ with size $M \times N$. After forming this data matrix, the frequently used statistical measures (mean and covariance) are calculated.

5.3.1.2 Compute the mean speech sample

The mean of the speech sample denotes central location of the whole data and is not necessarily equal to the middle values. The mean of the random vector $x$ is
computed from the Eq. 5.1

\[ m = E[x] \]  \hspace{1cm} (5.1)

where \( E[x] \) is the expected value of the argument \( x \). Where \( x \) is the random speech sample corresponding to column vector in the data matrix. The columns of the data matrix \( x(i) \) use the expression as shown in Eq 5.2

\[ m = \frac{1}{N} \sum_{i=1}^{N} x_i \]  \hspace{1cm} (5.2)

where \( N \) is the number of speech samples in the training data set and \( m \) represents the mean vector of the data matrix. It also represents the mean speech sample (when it is converted from column vector to matrix) in the training data set.

5.3.1.3 Compute the Covariance matrix

Covariance calculates the linear relationship between the two variables. So the covariance matrix has got two dimensions. If we have data samples with more than two dimensions, then we have many covariance values. For instance, if we have \( n \) dimensional data sets, then we get different covariance values. For example if we have three dimensional data sets with dimensions \( x, y, z \), we compute the covariance of \( x, y \) and the covariance of \( y, z \) and the covariance of \( x, z \). The covariance matrix \( C \) is a matrix which contains each entry as a covariance value. High covariance value indicates the high redundancy and low covariance value indicates low redundancy. The covariance matrix \( C \) of the random vector \( x \) is computed by using the Eq. 5.3 or Eq. 5.4

\[ C = E[(x - m)(x - m)^T] \]  \hspace{1cm} (5.3)

\[ C = \frac{1}{N} \sum_{i=1}^{N} (x_i - m)(x_i - m)^T \]  \hspace{1cm} (5.4)
\[ C = AA^T \] (5.5)

If we calculate the covariance matrix by using given Eq. 5.5, it utilizes high memory because of the dimensions of \( C \). The size of matrix \( A \) is \( M \times N \). The size of matrix \( C \) is \( M \times M \) which is very large. It is impractical to calculate \( C \) as shown in Eq. 5.6. Let us consider the matrix \( L \) instead of matrix \( C \).

\[ L = A^T A \] (5.6)

The dimension of matrix \( L \) is \( N \times N \) which is much smaller than the dimensions of matrix \( C \).

We can compute the eigenvalues and eigenvectors of covariance matrix. These eigenvectors and eigenvalues give the important information regarding the data samples. Eigenvectors provide the uncorrelated variables and are called principal components. The first principal component gives high amount of variation. The eigenvalues describes the variance of the corresponding principal component i.e. it gives that the first principal component which exhibits the highest amount of variation and the second principal component exhibits the second highest amount of variation and so on. Almost all appearance based techniques use statistical properties like mean and covariance to analyze the data. So we consider the highest significant eigen vectors (principal components) and leave the less significant eigenvectors.

### 5.3.1.4 Feature vector formation

After computing the eigenvectors of the covariance matrix, reduction in the dimensionality takes place. We do not consider all the eigenvectors as principal components. First we arrange all eigenvalues in the descending order and considers first few highest eigenvalues and corresponding eigenvectors. These eigenvectors e1, e2, ... etc., are the principal components and are shown in Eq. 5.7
\[ W = [e_1, e_2, ..., e_n] \] \hspace{1cm} (5.7)

The remaining less significant eigenvalues and corresponding eigen vectors are ignored. These ignored eigenvalues represent a very small information loss. The principal component axis pass through the mean values. With these principal components (eigenvectors), a feature vector matrix is formed (also called eigen space). A new transformation matrix \( W \) is calculated by projecting the principal component on to the original data set. Then this data set is formed with new representations in the feature space [129].

### 5.3.1.5 Derivation of a new data set

To derive a new data set with reduced dimensionality, we consider the transposition of the feature vector matrix (now each row of the matrix represents the eigenvector) and project this matrix on to the original data set with the subtracted mean. This will form a new transformation matrix by using the linear transformation of the original space into new reduced dimensional feature space by using Eq. 5.8

\[ Y = W^TX \] \hspace{1cm} (5.8)

The mean of the data matrix \( X \) is \( m \). In FLD, we have to calculate the mean of each class which is represented as \( m_i \) and \( i \) represents the specific class.

\[ S_w = \frac{1}{N} \sum_{i=1}^{c} N_i S_i \] \hspace{1cm} (5.9)

Within class scatter matrix is calculated by using the Eq. 5.9 \( S_w \) is the amount of variance between the speech samples in each class. \( N \) represents the sample vectors \( N_i \) is the number of samples in class \( i \) and \( S_i \) is the sum of the covariance matrix of the samples in each class. \( S_i \) is calculated by using the
Eq. (5.10)

\[
S_i = \frac{1}{N_i} \sum_{x \in X_i} (x - m_i)(x - m_i)^T
\]  

(5.10)

where

\[
m_i = \frac{1}{N_i} \sum_{x \in X_i} x_i
\]  

(5.11)

\[S_i\] represents the class dependent scatter matrix. \(X_i\) represents the data matrix corresponding to class \(i\). \(N_i\) represents the sample vectors present in class \(i\). \(c\) represents the total number of classes. The between class scatter matrix is given by the Eq. (5.12)

\[
S_b = \sum_{i=1}^{c} (m_i - m)(m_i - m)^T
\]  

(5.12)

where

\[
m = \frac{1}{N} \sum_{i=1}^{N} x_i
\]  

(5.13)

The overall scattering matrix is calculated by using the covariance matrix of all speech samples as shown in Eq. (5.14)

\[
S_m = E[(X - m)(X - m)^T] = S_w + S_b
\]  

(5.14)

If \(S_w\) is nonsingular, we should solve the generalized eigen problem for transformation matrix \(W\). This matrix \(W\) should minimize the within class scatter matrix and maximize the between class scatter matrix.

\[
S_b W = S_w W \lambda
\]  

(5.15)

If the transform \(W = [w_1, w_2, ..., w_m]\) is applied, the average within-class variation will be
\[ \tilde{S}_w = \frac{1}{N} \sum_{i=1}^{c} \left\{ \frac{1}{N_i} \sum_{i=1}^{c} \left( W^T x_i - \frac{1}{N_i} \sum_{i=1}^{c} (W^T x_i) \right) \left( W^T x_i - \frac{1}{N_i} \sum_{i=1}^{c} (W^T x_i) \right)^T \right\} \]  
\[ = w^T \left\{ \frac{1}{N} \sum_{i=1}^{c} cN_i S_i \right\} W \]  
\[ = w^T S_w W \]  
\[ (5.16) \]

In a similar way, the average between-class variation is given as

\[ \tilde{S}_b = w^T S_b W \]  
\[ (5.19) \]

There are many solutions to solve the generalized eigenvalue problem. After solving this, the result is

\[ J(W) = \frac{\tilde{S}_b}{S_w} = \frac{|w^T S_b W|}{|w^T S_w W|} \]  
\[ (5.20) \]

which is called as Fisher’s criterion \[49\]. All these steps are shown in the Fig. 5.1.

\[ \tilde{W} = \arg \max_w J(W) = \arg \max_w \frac{|\tilde{S}_b|}{|S_w|} = \arg \max_w |w^T S_b W| |w^T S_w W| \]  
\[ (5.21) \]

Or, for each column vector \( W_i \) of transformation matrix \( W \). The quadratic form has optimal solution \( \lambda_i = \frac{W_i^T S_b W_i}{W_i^T S_w W_i} \)

By applying the derivatives on both sides we get

\[ \Rightarrow \frac{\partial \lambda_i}{\partial W_i} = \frac{2S_b W_i (W_i^T S_w W_i) - 2S_w W_i (W_i^T S_b W_i)}{(W_i^T S_w W_i)^2} = 0 \]  
\[ (5.22) \]
\[ \Rightarrow \frac{S_b W_i (W_i^T S_w W_i)}{(W_i^T S_w W_i)^2} - \frac{S_w W_i (W_i^T S_b W_i)}{(W_i^T S_w W_i)^2} = 0 \] (5.23)

\[ \Rightarrow \frac{S_b W_i}{(W_i^T S_w W_i)} - \frac{S_w W_i}{(W_i^T S_w W_i)} \lambda_i = 0 \left( \lambda_i = \frac{W_i^T S_b W_i}{W_i^T S_w W_i} \right) \] (5.24)

\[ S_b W_i - S_w W_i \lambda_i = 0 \Rightarrow S_b W_i = S_w W_i \lambda_i \] (5.25)

One method of solving this eigen problem is to consider the inverse of 
\( S_w \) and solve the problem by using matrix \( S_w^{-1} S_b \), where \( \lambda \) is a diagonal matrix

\[ S_w^{-1} S_b = W \lambda \] (5.26)

To get the transformation matrix \( W \) we must calculate the eigenvalues
and eigenvectors by using the singular value decomposition of \( S_w^{-1} S_b \). This algo-

rithm is optimal when the scatter matrix is non singular only. If \( S_w \) is singular then
we get a warning message that the matrix is close to singular or badly scaled. This
is singularity problem and thus occurs due to high dimensional and low sample
size data.

### 5.3.2 Regularized Discriminant Analysis (RDA)

Before going to discuss the procedure for Regularized Discriminant Analysis, this
section will discuss the methods for solving the singularity problem that occur in
Linear Discriminant Analysis.

#### 5.3.2.1 Methods to solve Singularity problem

In general there exists 3 methods to solve the singularity problem
5.3.2.1.1 Subspace method

In this method initially PCA (Principle Component Analysis) is applied, an intermediate dimensionality reduction step, to reduce the dimensionality before LDA is applied. This algorithm is known as PCA+LDA, or subspace LDA. In this two-stage PCA+LDA algorithm, the discriminant stage is preceded by a dimensionality reduction stage done by using PCA. The dimensionality of the subspace transformed by PCA is chosen in a way such that the reduced total scatter matrix in this subspace is nonsingular, so that the classical LDA can be applied.

5.3.2.1.2 Regularization technique

The algorithm is known as Regularized Discriminant Analysis, or RDA in short. The key idea behind this is to add a constant \( \lambda \) to the diagonal elements of within-class scatter matrix \( S_w \). The elimination of the singularity problem depends on choosing the value of the regularized parameter \( \lambda \). In this thesis we concentrate more on this method in order to eliminate the singularity problem.

5.3.2.1.3 The null space LDA (NLDA)

In this NLDA, the ‘between-class’ distance is maximized in the null space of the ‘within-class scatter’ matrix. The singularity problem is thus eliminated implicitly by first removing the null space of the total scatter matrix. With this, the efficiency of the algorithm is improved. It is based on the observation that the null space of the total scatter matrix is the intersection of the null spaces of the between-class scatter matrix and within-class scatter matrix.

The procedure for solving the singularity problem, by using the Regularized Discriminant Analysis technique is discussed in the next section.
5.3.2.2 Procedure for Regularized Discriminant Analysis

The procedure for Regularized Discriminant Analysis is an extension of Linear Discriminant Analysis which eliminates the drawbacks that occur in LDA. The block diagram of Regularized discriminant Analysis is shown in the Fig. 5.2. If the covariance matrix $S_w$ is non singular then it comes under Linear Discriminant Analysis else it comes under Regularized Discriminant Analysis.

The main idea behind this Regularized Discriminant Analysis is to add a constant $\lambda$ to the diagonal elements of within class scatter matrix $S_w$ of the speech samples which is given in the Equation 5.27.
\[ S_w = S_w + \lambda I \]  

(5.27)

where \( \lambda \) is the regularized parameter which is relatively small in such a way that \( S_w \) is a positive definite. In this thesis work, the value of \( \lambda \) considered is \( 10^{-3} \). The estimation of regularization parameter value is very difficult as lower values of \( \lambda \) will not solve the singularity problem and higher values of \( \lambda \) will disturb the information in the within class scatter matrix in LDA[129][145].

The transformation matrix \( W \) is computed by using within class and between class scatter matrices \( S_w \) and \( S_b \) which is shown in the Eq. 5.28

\[ S_w^{-1} S_b W = W \lambda \]  

(5.28)

Once the transformation matrix \( W \) is given, the speech samples are projected on this matrix \( W \). After the projection, the distance between each train speech sample and the test speech sample is calculated by using the Euclidian distance measure, the minimum value among them will classify the result[60]. All these steps are explained clearly in the Fig 5.2.

5.3.3 Support Vector Machine(SVM)

SVM is an effective approach used for binary classification. This can be extended for multi class classification of speech samples by using a combination of binary class support vector machines. The basics of the Support Vector Machine can be found in references [132][12][21][54][42]. There are many ways for multi category SVM in the literature, among them ‘one-against-all’ and ‘one-against-one’ are most popular. Most of the people used one against all approach for classification of multi class problem.

A set of features are extracted from the speech sample like prosody and spectral features and are used as input for training the SVM classifier. MFCC,
Pitch and Energy are the extracted prosody and spectral features from the speech samples of different emotional classes. All these values are combined in the form of a matrix. Each column vector corresponds to values of different features and each row of the matrix is a representation of speech sample.

The input feature vectors in the feature space are transformed to a higher dimensional feature space. Then, the equation of the hyperplane which optimally separates the training speech samples with a maximum margin is calculated. To construct an optimal hyperplane one has to take into account of support vectors, which determines the margin [74]. As higher the margin being the more is the speech sample classification performance and vice versa. The block diagram of the support vectors are shown in the Fig. 5.3.

\[ (w \cdot x) + b = 0 \] (5.29)

where \( w \), the weight vector, \( x \), our row vector of corresponding speech
sample and $b$ is the bias. The distance from the hyperplane to the closest points (support vectors) of the two classes on both sides of the hyperplane is called margin $M$ of the hyperplane. To maximize the margin $M$, one has to minimize the subject to conditions.

$$\min_{w,b} \frac{1}{2} \|w\|^2 \text{ subject to } y_i((w.x_i) + b) \geq 1, \ i = 1, 2, \ldots n. \quad (5.30)$$

This is called quadratic optimization problem. Lagrangian function is used to solve the problem with appropriate Lagrange multipliers ($\alpha_i\alpha_k$).

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{n} \alpha_i(y_i((w.x_i) + b) - 1) \quad (5.31)$$

The Lagrangian $L$ has to be maximized with respect to the dual variables and minimized with respect to the primal variables $w$ and $b$. Then we obtain weight vector for the optimal hyperplane and is the linear combination of support vectors.

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (5.32)$$

where $y_i x_i$ are called support vectors, $y_i$ represents class label and $x_i$ represents the training speech sample. Here we trained 4 classes of emotions (happy, neutral, anger and sad) so we get 4 models one for each emotion. In each model we have $\alpha_i$ and bias $b$ values and support vectors $y_i x_i$. By using this support vectors, we can classify our test speech sample by using the decision function given in Eq. $5.33$.

$$f(x) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i x_i x + b) \quad (5.33)$$

The dot product is applied between each test speech sample with support
vectors, alpha and bias values which are obtained during the training phase. The data is transformed from input space to high dimensional feature space, with the mapping function \( \Phi : \mathbb{R}^n \rightarrow H \). After obtaining support vectors a linear kernel function is applied as shown in Eq. 5.34.

\[
f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right)
\]

(5.34)

where \( K(x_i, x) = x_i \cdot x_j \) is the kernel function. As a result we obtain several models in training phase which is equal to number of emotional classes chosen. These models and the generated support vectors will classify the emotion of the test speech sample.

### 5.3.4 k-Nearest Neighbor(kNN)

k Nearest Neighbor(kNN) is a non-parametric method for classifying the speech samples based on closest training samples in the feature space[95]. Similar to the RDA and LDA the speech samples are given as input to the kNN classifier. k Nearest Neighbor classification algorithm uses the training samples directly rather than that of a model derived in training from those samples. It represents each speech sample in a \( n - \text{dimensional space} \) where \( n \) is the number of features. The similarity of the test samples with the training samples is compared with Euclidian distance measure. Once a list of nearest neighbor speech samples is obtained, the input speech sample is classified based on the majority class of nearest neighbors.

\[
y' = \arg \max_v \sum_{(x_i, y_i) \in D_x} I(v = y_i)
\]

(5.35)

where
Each neighbour has the same impact on the classification. One way to diminish the impact is to weight the neighbour according to its distance.

\[ w_i = \frac{1}{d(y', y_i)^2} \]  

(5.36)

Training examples located far away are the lower impact on the classification. Because of this reason the classification can be done based on the Eq. 5.37.

\[ y' = \arg \max_v \sum_{(x_i, y_i) \in D} w_i * I(v = y_i) \]  

(5.37)

5.4 Distance Measures

Based on the literature survey, there exists several linear distance measuring techniques. Moon and Phillips explains different distance measures like Euclidean distance, Mahalanobis distance, Minkowski distance, city block and cosine distance metric. Let us consider a training set of \( n \) samples. Then compute the feature vector \( y \) with these samples i.e. feature vector \( y \) has \( n \) number of \((k \times 1)\) column vectors as \( y_1, y_2, ... y_n \). The feature vector of test sample is \( y_{tst} \). Let us compute the distance \( d \) between \( y_i \) and \( y_{tst} \) by using various distance measures,
where $i$ represents the $i^{th}$ column vector.

$$ y_{(i=1,2,...,k)} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & \cdots & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k1} & \cdots & \cdots & y_{kn} \end{pmatrix} \tag{5.38} $$

### 5.4.1 Euclidean distance

The most commonly used distance measure in many applications is the Euclidean distance and is a straight line distance. This gives the shortest distance between the two samples or vectors and is similar to the Pythagorean equation in two dimensions \[87\]. The sum of squared distance of two feature vectors $(y_{tst}, y_i)$ is given by Eq. 5.39

$$ d^2 = (y_{tst} - y_i)^T (y_{tst} - y_i) \tag{5.39} $$

$$ d = \sum_{j=1}^{n} |y_{tstj} - y_{ij}| \tag{5.40} $$

The Euclidean distance is very much sensitive in both adding and multiplying the vector with some factor or value.

### 5.4.2 Standardized Euclidean distance

The Equation for standardized Euclidean distance measure is given in Eq. 5.41

$$ d^2 = (y_{tst} - y_i)^T D^{-1} (y_{tst} - y_i) \tag{5.41} $$

For this distance measure takes the variance $v_j^2$ of the $j^{th}$ element of $y$ whose observations are the $y_i$’s, $(i = 1, 2, .., N)$ where $D$ is the diagonal matrix with
diagonal elements containing the variance $v_j^2$. The standardized Euclidean distance measure does not improve the recognition rate as compared with the Euclidean distance measure. If we use the standardized Euclidean distance measure, the number of eigenvectors increase, resulting in decrease in recognition rate.

5.4.3 Mahalanobis distance

Mahalanobis distance measure comes from the Gaussian multivariate probability density function as defined in Eq. 5.42

$$p(x) = (2\pi)^{-d/2}|C|^{-1/2} \exp\left(-1/2(x - m)^T C^{-1} (x - m)\right) \tag{5.42}$$

where $(x - m)^T C^{-1} (x - m)$ is called squared Mahalanobis distance, which is important in characterizing the distribution where $C$ is the estimated covariance matrix of $y$ whose observations are $y_i$’s s. The Mahalanobis distance of two feature vectors $y_{tst}$ and $y_i$ is computed using the Eq. 5.43

$$d^2 = (y_{tst} - y_i)^T D^{-1} (y_{tst} - y_i) \tag{5.43}$$

Both the Mahalanobis distance measure and the standardized Euclidean distance measure will give almost all the same performance.

5.4.4 Cityblock distance

The city block distance measure is defined by the Eq. 5.44

$$d = \sum_{j=1}^{n} (y_{tstj} - y_{ij}) \tag{5.44}$$

where $y_{tstj}$ is the test sample row vector and $y_{ij}$ is the row vector of a trained matrix.
5.4.5 Minkowski distance

\[ d = \left( \sum_{j=1}^{n} |(y_{tstj} - y_{ij})|^p \right)^{1/p} \] (5.45)

Minkowski distance measure is the special case of the city block distance when \( p = 1 \) where as the Euclidean distance measure is the special case of Minkowski distance measure when \( p = 2 \).

5.4.6 Cosine distance

The Cosine distance measure is defined by using the Eq. 5.46

\[ d = (1 - \frac{y_{tst}^T y_i}{(y_{tst}^T y_{tst})^{1/2}(y_i^T y_i)^{1/2}}) \] (5.46)

The Cosine and the Euclidean distance measures give almost equal accuracy \([73]\). When compared to all the given distance measures, the Euclidean distance measure gives the better performance. So Euclidean distance measure is used in most of the applications. Wendy et al claims that the mahalanobis distance measure proposed by Moon’s \([65]\) gives better performance when compared to the Euclidian distance measure, which is given in Eq. 5.47

\[ d = -\sum_{j=1}^{n} \frac{\lambda_j}{\lambda_j + \alpha^2} y_{tst,j} y_{i,j} \] (5.47)

consider \( \alpha = 0.25 \)

\[ \frac{\lambda_j}{\lambda_j + \alpha^2} \approx \frac{1}{\sqrt{\lambda_j}} \] (5.48)

Then distance measure formula is modified and is given as

\[ d = -\sum_{j=1}^{n} 1 \sqrt{\lambda_j} y_{tst,j} y_{i,j} \] (5.49)
Where $\lambda_j$ is the $j^{th}$ eigenvalues corresponding to $j^{th}$ eigenvector.

5.4.7 Hausdorff distance

Hausdorff distance is a special distance metric which is used to calculate the distance between different sized matrices having a single common dimension, like the acoustic matrices representing our speech feature vectors. It derives from the Hausdorff metric for sets \[28\][9].

This will measures the extent to which each point of a ‘model’ set lies near some point of an sample set and vice versa. Unlike the most vector comparison methods, the Hausdorff distance is not based on searching corresponding mode and sample points \[119\]. Thus, it is more tolerant of perturbations in the location of points because it computes proximity rather than the exact superposition.

However, the Hausdorff distance measure is extremely sensitive to outliers. If two sets of points $A(a_1, a_2)$ and $B(b_1, b_2, b_3)$ are similar, then all points are perfectly superimposed except one single point in A which is far from any point in B, then the Hausdorff distance is determined by that point and is large. This sensitivity is not acceptable for outliers. So a few modified Hausdorff distances and directed Hausdorff distance are used for the voice recognition.

5.4.8 The directed Hausdorff distance

The distance between two points $a$ and $b$ is given as $d(a, b) = \| a - b \|$. Here, we calculate not only the distance between the point $a$ in the finite point set $A$ and the same value in the finite point set $B = b_1, ... b_{N_b}$ but also calculating the distances between the $a_t$ and its two neighbor values $b_{t-1}$ and $b_{t+1}$ in the finite point set $B$ respectively, then minimize these three distances \[9\].

$$d(a, B) = \min_{b \in B} d(a, b) = \min_{b \in B} \| a - b \|$$

(5.50)
The directed Hausdorff distance metric $h(A,B)$ between the two finite point set \( A = a_1, \ldots a_{N_A} \) and \( B = b_1, \ldots b_{N_B} \) is defined as follows

$$ h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b) = \max_{a \in A} \min_{b \in B} \| a - b \| $$  \hspace{1cm} (5.51)

$$ h(A, B) = \left\{ \max_{a \in A} \left\{ \min_{b \in B} \| a - b \| \right\} \right\} $$ \hspace{1cm} (5.52)

In this way, \( h(B, A) \) is

$$ h(B, A) = \max_{b \in B} \min_{a \in A} d(b, a) = \max_{b \in B} \min_{a \in A} \| b - a \| $$  \hspace{1cm} (5.53)

$$ h(B, A) = \left\{ \max_{b \in B} \left\{ \min_{a \in A} \| b - a \| \right\} \right\} $$ \hspace{1cm} (5.54)

where \( d \) is a proper metric between the points of sets \( A \) and \( B \) (for instance, the Euclidean distance). It is named also as forward Hausdorff distance, while \( h(B, A) \) represents the backward Hausdorff distance for sets. Thus, we can obtain the general definition for the Hausdorff distance between sets as follows

$$ h(A, B) = \max \{ h(A, B), h(B, A) \} $$ \hspace{1cm} (5.55)

from the Eq. 5.52, Eq. 5.54, Eq. 5.55, the next Hausdorff distance formula is obtained as follows

$$ h(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right\} $$ \hspace{1cm} (5.56)

The components of this Hausdorff distance, are sometimes named as forward and backward Hausdorff distances from \( A \) to \( B \) or vice versa. Let us consider the matrices having a single common dimension (the number of rows), instead of sets. Thus \( A = (a_{ij})_{n \times m} \) and \( B = (b_{ij})_{n \times p} \). Let us introduce two more helping vectors or auxiliary vectors, \( y = (y_i)_{p \times 1} \) and \( z = (z_i)_{m \times 1} \), then calculate \( || y ||_p = \max_{0 \leq i \leq p} || y_i || \) and \( || z ||_m = \max_{0 \leq i \leq m} || z_i || \) respectively. With
these notations a new nonlinear metric $d$ has been generated which is having the following form

$$
d(A, B) = \max \left\{ \sup_{\|y\|_p \leq 1} \inf_{\|z\|_m \leq 1} \| B_y - A_z \|, \sup_{\|y\|_p \leq 1} \inf_{\|z\|_m \leq 1} \| B_y - A_z \| \right\} \quad (5.57)
$$

This is restriction based metric which represents the Hausdorff distance between the sets $B(y : \| y \|_p \leq 1)$ and $A(\| z \|_m \leq 1)$ in the metric space $\mathbb{R}^n$. Therefore it can be obtained using the following equation

$$
d(A, B) = H(B(y : \| y \|_p \leq 1), A(\| z \|_m \leq 1)) \quad (5.58)
$$

From the Eq. (5.58) we have

$$
B_y - A_z = \sum_{k=1}^p b_{ik} y_k - \sum_{j=1}^m a_{ij} z_j \quad (5.59)
$$

$$
\| B_y - A_z \|_n = \max_{1 \leq i \leq n} \| \sum_{k=1}^p b_{ik} y_k - \sum_{j=1}^m a_{ij} z_j \| \quad (5.60)
$$

There by, deducing the following equation

$$
\sup_{\|y\|_p \leq 1 \|z\|_m \leq 1} \inf_{\|y\|_p \leq 1 \|z\|_m \leq 1} \| B_y - A_z \|_n = \sup_{\|y\|_p \leq 1 \|z\|_m \leq 1} \inf_{\|y\|_p \leq 1 \|z\|_m \leq 1} \max_{1 \leq i \leq n} \| \sum_{k=1}^p b_{ik} y_k - \sum_{j=1}^m a_{ij} z_j \| \quad (5.61)
$$

This can be observed as a max min optimization problem and according to the classical J.von Neumann min max theorem \cite{9} we obtain

$$
\sup_{\|y\|_p \leq 1 \|z\|_m \leq 1} \inf_{\|y\|_p \leq 1 \|z\|_m \leq 1} \| B_y - A_z \|_n = \inf_{\|z\|_m \leq 1} \sup_{\|y\|_p \leq 1} \| B_y - A_z \|_n \quad (5.62)
$$

After eliminating the terms $y$ and $z$ from the above formula, we finally
get the following Hausdorff based distance measure

\[ d(A, B) = \max \left\{ \sup_{1 \leq i \leq m} \inf_{1 \leq k \leq p} |b_{ik} - a_{ij}|, \sup_{1 \leq i \leq m} \inf_{1 \leq k \leq p} |b_{ik} - a_{ij}| \right\} \quad (5.63) \]

The nonlinear function \( d \) thus arrived verifies the main distance properties:

- Non-negativity: \( d(A, B) \geq 0 \)
- Symmetry: \( d(A, B) = d(B, A) \)
- Triangular inequality: \( d(A, B) + d(B, C) \geq d(A, C) \)

without representing a Hausdorff metric no longer. The Hausdorff-based distance \( d \) represented by the Eq. \( 5.63 \) constitutes a satisfactory discriminator between the vocal feature vectors \( [48] \). It defines a new metric topology on the space of all matrices \( \{A\} \), that is not same but comparable with that induced by Hausdorff topology. The newly introduced distance measure can be successfully used in the voice feature vector classification process.

5.4.8.1 Example

Initially, we find the distance between \( a_1 \) and \( b_j' \), \( a_2 \) and \( b_j' \) and saving the shortest distances. Finally, we find the larger distance among the two shortest distances \( a_1 \) and \( b_j' \), \( a_2 \) and \( b_j' \).

From the Fig. 5.4, we calculate the distance between \( a_1 \) and \( b_j' \) and keep the shortest distance, similarly \( a_2 \) and \( b_j' \) and keep the shortest distance. Finally, we get the largest of the two distances from the Fig. 5.5 5.6. Basing on this, we can say that any point of A is at utmost distance \( h(A, B) = d(a_1, b_1) \) to some point of B.
CHAPTER 5. RELATED PATTERN RECOGNITION BASICS 88

Figure 5.4: Given two sets of points A and B, find h(A,B)

Figure 5.5: To compute the distance between \( a_1 \) and \( b_j \)'s (b) shortest distance \( a_1 \) to \( b_1 \)

Figure 5.6: To compute the distance between \( a_2 \) and \( b_j \)'s (b) shortest distance \( a_2 \) to \( b_3 \)

Figure 5.7: To Find the largest among the two distances from \( a_1 \) and \( a_2 \) (b) largest distance among those two shortest distances
5.5 Summary

This chapter has discussed about pattern recognition basics and explained various pattern recognition approaches. Those approaches are Statistical, Syntactical, Template matching and Neural Network approaches. The statistical approach based classification techniques (LDA, RDA, SVM and KNN) are implemented for this work and are discussed here in theoretical perspective. Various types of distance measure techniques are also explained and compared with. In these distance measures, Euclidian distance is used for this work, which gives the better performance.