DIMENSIONALITY REDUCTION USING
FACTOR ANALYSIS

A Dissertation Submitted in Fulfillment of the
Requirements of the Degree of
Master of Philosophy

by

NITIN KHOSLA
B.E.(Electronics), M.E.(Computer Science)

School of Microelectronics Engineering
Griffith University, Australia

December, 2004
Abstract

In many pattern recognition applications, a large number of features are extracted in order to ensure an accurate classification of unknown classes. One way to solve the problems of high dimensions is to first reduce the dimensionality of the data to a manageable size, keeping as much of the original information as possible and then feed the reduced-dimensional data into a pattern recognition system. In this situation, dimensionality reduction process becomes the pre-processing stage of the pattern recognition system. In addition to this, probability density estimation, with fewer variables is a simpler approach for dimensionality reduction. Dimensionality reduction is useful in speech recognition, data compression, visualization and exploratory data analysis.

Some of the techniques which can be used for dimensionality reduction are - Factor Analysis (FA), Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Factor Analysis can be considered as an extension of Principal Component Analysis. The EM (expectation maximization) algorithm is ideally suited to problems of this sort, in that it produces maximum-likelihood (ML) estimates of parameters when there is a many-to-one mapping from an underlying distribution to the distribution governing the observation. conditioned upon the obervations. The maximization step then provides a new estimate of the parameters.

This research work compares the techniques - Factor Analysis (Expectation-Maximization algorithm based), Principal Component Analysis and Linear Discriminant Analysis for dimensionality reduction and investigates Local Factor Analysis (EM algorithm bassed) and Local Principal Component Analysis using Vector Quantization.
I would like to express my deep and sincere gratitude to my supervisor Professor Kuldip K. Paliwal for his valuable suggestions, continuous guidance and encouragement during all the phases of the thesis period. Without his support this thesis would not have been materialized.

My sincere thanks to my fellow researchers in the signal processing laboratory of the school for their comments, suggestions and help.

I have no words to express my gratitude and thanks to my father, S.K. Khosla, my wife, Jaya and my daughters, Surabhi and Jaiti for their love, constant inspiration and moral support all through the time of my studies.

NITIN KHOSLA
B.E.(Electronics), M.E.(Computer Science)
Statement of Originality

This work has not been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.
# Contents

## Chapter 1: Introduction

1.1 Motivation ................................................................................... 1  
1.2 Problem Formulation ................................................................. 2  
1.3 Scope of Research ...................................................................... 4  
1.4 Contribution Resulting From This Thesis ................................. 4  
1.5 Thesis Outline .......................................................................... 5  

## Chapter 2: Pattern Classification

2.1 Pattern ....................................................................................... 6  
2.2 Class ........................................................................................... 7  
2.3 Simple Classifier ........................................................................ 7  
2.4 Pattern Recognition System Design Steps ............................... 8  
2.5 Minimum Distance Classifier .................................................... 9  
2.6 Mahalanobis Classifier .............................................................. 11  
2.7 Maximum-likelihood Parameter Estimation .............................. 12  

## Chapter 3: Dimensionality Reduction Methods

3.1 Introduction ............................................................................... 14  
3.2 Problem Definition of Dimensional Reduction ......................... 16  

3.2.1 The Curse of the Dimensionality ......................................... 17  
3.2.2 Intrinsic Dimension of a Sample ........................................... 19  
3.3 Dimensionality Reduction Methods ......................................... 20  
3.4 F-Ratio ....................................................................................... 21  
3.5 Recognition Rate on Training Data ......................................... 22  
3.6 Linear Discriminant Analysis .................................................... 22  
3.7 Principal Component Analysis .................................................. 23  

3.7.1 Geometric Representation of Principal Components ....... 24  
3.8 Factor Analysis .......................................................................... 29
List of Figures

1.1 Transforming High Dimensional Data to Low Dimension .......... 3
2.1 A Simple Classifier ............................................................... 8
2.2 Basic Steps of a Pattern Recognition System Design Process ..... 8
2.3 Minimum Distance Classifier Block Diagram ......................... 10
2.4 Contours of Various Distance Measures .............................. 10
2.5 Mahalanobis Distance Classifier Block Diagram ..................... 11
3.1 Dimensionality Reduction Mappings and the Reconstruction Mappings ................................................................. 18
3.2 Scatter plot of two standardized variables .............................. 25
3.3 1-D Representation by Largest Principal Components of 2-D Data .. 26
3.4 Distribution of Two Uncorrelated Variables ............................ 27
3.5 Regression Line and Principal Component ............................. 28
3.6 Largest Principal Component of Covariance Matrix .................. 28
5.1 VQ Codebook Formation ..................................................... 54
5.2 Flow-chart for LBG-VQ Algorithm ...................................... 56
The amount of computations required for pattern recognition and the amount of data required for training systems grow exponentially with the increase of the dimensionality of the feature vectors. It is often required for adding new features if the performance of the pattern recognition system is inadequate. Though addition of new features improves the pattern recognition performance, yet it creates some problems as described in [2] -

(a) The recognizer using a large feature set is computationally more complex and requires more storage which makes its real-time implementation very difficult and costly.

(b) A large amount of data is required for training the recognizer if the size of the feature set is increased.

1.1 MOTIVATION

**Dimensionality Reduction** is one of the key techniques in pattern recognition, aimed at revealing meaningful structures and unexpected relationships in the multivariate data. It assembles numerous methods, all striving to present high dimensional data in a low dimensional space, in a way that faithfully captures desired structural elements of the data[65][78].

Dimensionality reduction is used for many purposes. It is beneficial as a visualization tool to present multivariate data in a human accessible form, as a method
of feature extraction in speech recognition, and as a preliminary transformation applied to the data prior to the use of analysis tools in classification[75][76].

Dimensionality reduction techniques possess several significant advantages. Some of them, as described in [2][67][74][76], are -

1. The low dimensional data is reliable in the sense that it is guaranteed to show genuine properties of the original data.

2. Computational complexity of low dimensional data is low, both in time and space.

1.2 PROBLEM FORMULATION

Dimensionality reduction is an important part of a pattern recognition system. It is a process in which we represent a system having many degrees of freedom by a smaller number of degrees of freedom.

The main aim of the dimensionality reduction algorithms is to obtain a compact, accurate representation of the data that reduces or eliminates statistically redundant components.

Factor Analysis, Principal Component Analysis and Linear Discriminant Analysis are some of the techniques, which can be used for dimensionality reduction. Factor analysis can be considered as an extension of principal component analysis[73]. The approximation based on the factor analysis model is more elaborate than that of principal component analysis[73]. These techniques can be used to obtain a parsimonious description of the multi-variate data.

The EM (expectation maximization) algorithm is ideally suited to problems of this sort, in that it produces maximum-likelihood (ML) estimates of parameters when there is a many-to-one mapping from an underlying distribution to the distribution governing the observation. The EM algorithm consists of two steps: an expectation step, followed by a maximization step. The expectation ia with respect to the unknown underlying variables, using the current estimate of the parameters and conditioned upon the observations. The maximization step then provides a new estimate of the parameters.

Factor Analysis consists of the procedures for analysing the relations among a set of random variables observed or counted or measured for each individual of a group.
The purpose of a factor analysis is to account for the intercorrelations among \( n \) variables, by postulating a set of common factors, considerably fewer in number than the number, \( n \), of these variables. The factors are random variables that cannot be observed, counted or measured directly, but which are presumed to exist in the population and hence they are in the experimental sample[65][67][73].

**Linear Discriminant Analysis** is a method in which dimensionality reduction is done by projecting the original \( D \)-dimensional space on a \( d \)-dimensional subspace, where \( d < D \) and finding a linear transformation that defines this sub-space using the LDA. It is assumed that the within class distributions are Gaussian and the within-class covariance matrices are equal[2][38][39].

In **Principal Component Analysis**, the dimensionality reduction is achieved by projecting the original \( D \)-dimensional feature space on a \( d \)-dimensional subspace and finding the orientation of the subspace which best preserves the information available in the original space. The input pattern in the original \( D \)-dimensional feature space is transformed to the Karhunen-Loeve (KL) coordinate system and the dimensionality is reduced by representing the pattern by \( d \) coordinates in the KL coordinate system. The KL coordinate system represents optimally a set of \( D \)-dimensional patterns by another set of vectors of a lower dimensionality[38][39][53].

**Vector Quantization** can be used for implementing local factor analysis and local principal analysis. The VQ is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and it can be represented by the centroid called the codeword. The collection of all codewords consists of the corresponding codebook[17]. The application of dimensionality reduction techniques in each region in the VQ process may affect the pattern recognition performance[65][75][83].

![Fig 1.1 Transforming High Dimensional Data to Low Dimensional Data](image-url)
1.3 SCOPE OF RESEARCH

This research work compares the techniques - Factor Analysis (expectation-maximization based), Principal Component Analysis and Linear Discriminant Analysis for dimensionality reduction and investigate Local Factor Analysis (EM based) and Local Principal Component Analysis using Vector Quantization.

1.4 CONTRIBUTION RESULTING FROM THIS THESIS

(1) While comparing PCA, LDA and EM_FA for global dimensionality reduction on some standard machine learning databases, it is found that the recognition accuracy for LDA is higher than the other two in most of the cases. It is found that the recognition accuracy for LDA method is 98.67%, 51.97%, 85.28%, 97.73%, 80.11% and 94.50% for IRIS, PB, GLASS, WINE, BCW and Deterding datasets respectively.

(2) FA (EM based) and PCA were applied for local dimensionality reduction using vector quantization. The performances of VQPCA and VQ_EM_FA are compared for dimensionality reduction, and VQPCA is found performing better than VQ_EM_FA. When the number of partitions were 4, the recognition accuracy of VQPCA for IRIS, PB and Deterding datasets was 96%, 31.32% and 70.64% respectively. Whereas, in the case of VQ_EM_FA, the recognition accuracy was found to be 94.67%, 31.84% and 59.09% respectively.

(3) The local dimensionality reduction algorithms, VQPCA and VQ_EM_FA, partition the input space using the VQ and build PCA and EM_FA models respectively in each local region. Global models, such as PCA, FA, can incur a large error when data has different structure in different regions of the input space. Building simple linear models in local regions of the input space can be more accurate. After comparing FA with VQ-EM_FA and PCA with VQPCA, it has been found that VQFA and VQPCA perform better than FA and PCA, respectively, as the number of partitions (levels) of the input space is increased.
1.5 THESIS OUTLINE

Chapter-2 introduces some fundamentals of pattern classification. It discusses pattern recognition system design steps, minimum distance classifier, mahalanobis classifier.

Chapter-3 describes the concept of dimensionality reduction and some major techniques used for dimensionality reduction. The curse of dimensionality and intrinsic dimension of a sample are discussed. Dimensionality reduction methods such as F-ratio, LDA, PCA and Factor Analysis and linear transformation of data are described.

Chapter-4 covers various models and methods of factor analysis. It covers orthogonal factor model and deals with the methods of covariance structure estimation such as principal factor model and expectation-maximization based factor analysis method.

Chapter-5 discusses the issues related to local dimensionality reduction and its requirements. Vector Quantization, VQPCA and VQ_EM_FA are also described in detail.

Chapter-6 consists of various experiments performed with factor analysis, principal component analysis, linear discriminant analysis for dimensionality reduction and also experiments for local dimensionality reduction using vector quantization on some standard databases.

Chapter-7 describes the conclusion and future research work.
Pattern recognition deals with technical and mathematical aspects of classifying different objects through their observable information, such as pixels of an image or energy levels in frequency domain for a waveform. Patterns can be represented in many forms - visual patterns, temporal patterns, etc. Pattern recognition can be considered as the classification problem, where the input is assigned to a category[39].

2.1 PATTERN

Pattern is a quantitative or structural description of an object or some other entity. It can be arranged in the form of a vector such as:

\[
x = \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
. \\
. \\
x_n
\end{bmatrix}
\]

where \(x_1, x_2, x_3, \ldots, x_n\) are the features. These features can be discrete numbers and they reflect the characteristics of the desired objects. Features of different object differ in large content.

We can consider \(x\) being a point in a \(n\)-dimensional feature space. By this process of feature measurement, we can represent an object as a point in feature space.
2.2 CLASS

Class is a set of patterns that share some common properties. Feature vectors of the same type of objects form a set. Patterns extracted from the same types of objects are seldom identical. They can be interpreted as clusters of points in a n-dimensional space, which are called the distributions of classes. The purpose of pattern classification is to classify these patterns and thus the distribution of classes should be separable[38].

2.3 SIMPLE CLASSIFIER

A simple classification model can be considered of two main stages -

- Feature Extractor
- Classifier

The feature extractor is used to determine the numerical values for a set of $d$
features $x_1, x_2, x_3, ..., x_n$, which comprises the components of a feature vector $\mathbf{x}$. The classifier receives this $\mathbf{x}$ and then assigns it to one of the categories, Class 1, Class 2, ... , Class n. Practically it is found that, the feature extractor will always produce different feature vectors for different inputs, but it is assumed that the within-class variability is very small as compared to between-class variability[38][39].

2.4 PATTERN RECOGNITION SYSTEM DESIGN STEPS

![Flowchart of Pattern Recognition System Design Process](image-url)
The design of a pattern recognition system consists of many activities - data collection, feature choice, model choice, training and testing. The choice of the distinguishing features is a critical design step and depends upon the characteristics of the problem definition. Prior knowledge also plays an important role in feature extraction and selecting a suitable model (Figure 2.2)[38]. It is desirable to find features that are simple to extract, invariant to irrelevant transformations, insensitive to noise and are useful for discriminating patterns in different categories.

The process of using data to determine the classifier is referred to as training the classifier. Testing is done to measure both the performance of the system and to identify the need for improvements in its components.

2.5 MINIMUM DISTANCE CLASSIFIERS

Template matching is a common approach to pattern classification. Template matching can be expressed mathematically. Let $\mathbf{x}$ be the feature vector for the unknown input, and let $\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_n$ be templates (i.e., perfect, noise-free feature vectors) for the $n$ classes. Then the error in matching $\mathbf{x}$ against $\mathbf{m}_k$ is given by

$$||\mathbf{x} - \mathbf{m}_k||$$

A minimum-error classifier computes $||\mathbf{x} - \mathbf{m}_k||$ for $k = 1$ to $n$ and chooses the class for which this error is minimum. The term $||\mathbf{x} - \mathbf{m}_k||$ is also the distance from $\mathbf{x}$ to $\mathbf{m}_k$, it is called as a minimum-distance classifier[39].

$$||\mathbf{u}||$$ is called the norm of the vector $\mathbf{u}$. There are many ways to define the norm.
and these correspond to different ways of measuring distance[38]. Two of the most common ways of measuring distances are -

1. **Euclidean metric**: \( \| \mathbf{u} \| = \sqrt{u_1^2 + u_2^2 + \ldots + u_d^2} \)

2. **Manhattan metric** \( \| \mathbf{u} \| = |u_1| + |u_2| + \ldots + |u_d| \)

**Limitations of Minimum Distance Classifier**

It has been observed that many times this classifier gives errors. Some of the reasons for this include[65] -

1. The features may be highly correlated.
2. The feature set may be too complex.
3. The features may be inadequate to distinguish the different classes.
4. The decision boundaries are not curved.

5. There may be distinct sub-class in the data.

### 2.6 MAHALANOBIS CLASSIFIERS

Numerical value for a feature $x$ depends on the units used, i.e. on the scale. Sometimes it is desirable to scale the data so that the resulting standard deviation remain in unity[38][39]. The standardized distance from $x$ to $m$ is given by

$$ r = \frac{|x - m|}{s} $$

where $s$ is the standard deviation. Note that $r$ is invariant to translation and invariant to scale. This suggests an important generalization of a minimum-Euclidean-distance classifier. Let $x(i)$ be the value for Feature $i$, let $m(i,j)$ be the mean value of Feature $i$ for Class $j$, and let $s(i,j)$ be the standard deviation of Feature $i$ for Class $j$. In measuring the distance between the feature vector $x$ and the mean vector $m_j$ for Class $j$, suppose that we use the standardized distance[38][39]

$$ r(x, m_j)^2 = \left[ \frac{x(1) - m(1,j)}{s(1,j)} \right]^2 + \left[ \frac{x(2) - m(2,j)}{s(2,j)} \right]^2 + \ldots + \left[ \frac{x(d) - m(d,j)}{s(d,j)} \right]^2 $$

![Figure 2.5 Mahalanobis Distance Classifier Block Diagram (after[38])](image)
This distance has the important property that it is **scale invariant**. That is, if we measure distance in this way, the units we use for the various features will have no effect on the resulting distances, and thus no effect on the final classification. The Mahalanobis distance is a generalization of this standardized distance\[65]\.

The quantity $r$ is \[38\][39] 

$$ r_2 = (x - m_x)' C_x^{-1} (x - m_x) $$

called the Mahalanobis distance from the feature vector $x$ to the mean vector $m_x$, where $C_x$ is the covariance matrix for $x$. Mahalanobis distance can be used in a minimum-distance classifier. Let $m_1, m_2, ..., m_n$ be the means for the $n$ classes and let $C_1, C_2, ..., C_n$ be the corresponding covariance matrices. A feature vector $x$ can be classified by measuring the Mahalanobis distance from $x$ to each of the means, and assigning $x$ to the class for which the Mahalanobis distance is minimum.

The use of Mahalanobis distance is useful in many ways as compared to the Euclidean distance metric\[39][65] - 

1. It corrects the correlation between the different features.
2. It automatically accounts for the scaling of the coordinate axes.
3. It can provide curved as well as linear decision boundaries.

But the covariance matrices are hard to determine accurately. Moreover, as the number of features increases, this process requires more memory and time. These problems are generally insignificant when a few features are needed, but they can become quite serious when the number of features becomes large\[38][39].

### 2.7 MAXIMUM-LIKELIHOOD PARAMETER ESTIMATION

The problem of parameter estimation is classical one in statistics, and it can be approached in many ways. Maximum-likelihood estimation and Bayesian estimation methods are such two common procedures\[38].
Maximum likelihood method views parameters as quantities whose values are fixed but unknown. The best estimate of their value is defined to be the one that maximizes the probability of obtaining the samples actually observed. Bayesian methods view the parameters as random variables having some known prior distribution\[38\].

Maximum-likelihood estimation methods have many attractive properties. They nearly always have good convergence properties as the number of training samples increases. Maximum-likelihood estimation is often simpler than alternative parameter estimation methods.

For analytical purposes, it is usually easier to work with logarithm of the likelihood than the likelihood itself. Because the logarithm is monotonically increasing, the $\theta$ that maximizes the log-likelihood also maximizes the likelihood. The maximum-likelihood estimate of $\theta$ is, by definition, the value of $\theta$ that maximizes $p(D \mid \theta)$. This estimate corresponds to the value of $\theta$ that in some sense best agrees with or supports the actually observed training samples\[38\].
There are many applications in which a system processes data in the form of a collection of real-valued vectors—speech signals, images, etc. But the system is not very effective because the dimension of each individual vector—the number of components in the vector—is too high.

3.1 INTRODUCTION

The problem of dimensionality reduction appears when these data are in fact of a higher dimension than desirable (tolerable) limits. Let us consider the following typical cases:

(1) Let us assume that a pattern recognition system consists of 8000 dimensional vectors. If a multilayer perceptron is to be used as the classification system, the number of weights would be exceedingly large and would require an enormous training set to avoid overfitting. Therefore, the necessity of dimensionality reduction arises.[75]

(2) Let us consider a statistical analysis of a multivariate population. There can be a few variables and the analyst is interested in finding clusters of the population and/or interpreting the variables. To that aim, it is quite necessary to visualise the data, which requires reducing its dimensionality to 2 or 3 [75].

On a number of occasions it becomes necessary or useful to first reduce the dimensionality of the data to a manageable size, keeping as much of the original
information as possible. Then feed the reduced-dimension data into the whole system. In this situation, dimensionality reduction is the preprocessing stage in the whole system[75].

Whenever the intrinsic dimensionality of a data set is smaller than the actual one, dimensionality reduction can bring an improved understanding of the data apart from a computational advantage. Dimensionality reduction can be seen as a feature exraction or in general as a representation in a different coordinate system[75].

(1) **Hard Dimensionality Reduction Problems**: When the data have dimensions ranging from hundreds to several hundreds of thousands of components, then a drastic reduction is required. The components are often repeated measures of a certain magnitude in different points of space or in different instants of time. Pattern recognition and classification problems come in this category such as speech (e.g. auditory models, robust recognition systems), face recognition, character recognition systems, etc [75].

(2) **Soft Dimensionality Reduction Problems**: When the data is not too high dimensional, ranging around a few tens of components then the reduction is not very drastic. Generally the components are observed or measured values of different variables, which have a straightforward interpretation. The problems in the fields linked social sciences, psychology, etc come under this category[75].

(3) **Visualisation Problems**: When the data does not normally have a very high dimension in absolute terms, then we reduce it to 2 or 3 in order to plot it[75].

If we consider the time variable in, there can be another two categories: *static dimensionality reduction* and *time-dependent dimensionality reduction*. The later could be useful for vector time series, such as video sequences or continuous speech[75][65].

The amount of computations required for pattern recognition and the amount of data required for training systems grow exponentially with the increase of the dimensionality of the feature vectors. The addition of new features is often required if the performance of the pattern recognition system is inadequate[78]. Though the addition of new features improves the speech recognition performance, it also creates some problems such as[2][65] -
(a) The recognizer using the large feature set is computationally more complex and requires more storage which makes its real-time implementation very difficult and costly.

(b) A large amount of data is required for training the recognizer if the size of the feature set is increased.

In order to avoid these problems, the dimensionality of the enhanced feature set is to be reduced without affecting the recognition performance[27][75][78].

3.2 PROBLEM DEFINITION OF DIMENSIONALITY REDUCTION

Let there be a sample \( \{ \mathbf{t}_n \}_{n=1}^N \) of \( D \)-dimensional vectors lying in a data space \( T \) (usually \( \mathbb{R}^D \) or a subset of it). The assumption which justifies the dimensionality reduction is that the sample actually lies, at least approximately, on a manifold (nonlinear in general) of smaller dimension than the data space. The goal of dimensionality reduction is to find a representation of that manifold (a coordinate system) that will allow to project the data vectors on it and obtain a low-dimensional, compact representation of the data[75].

Given a sample \( \{ \mathbf{t}_n \}_{n=1}^N \subset T \), find:

1. A space \( \mathcal{P} \) of dimension \( L \) (typically \( \mathbb{R}^L \) or a subset of it)
2. A dimensionality reduction mapping \( \mathbf{F} : \mathcal{P} \rightarrow \mathcal{T} \)
3. A smooth, nonsingular, reconstruction mapping \( \mathbf{f} : \mathcal{P} \rightarrow M \subset \mathcal{T} \)

such that

(a) \( L < D \) is as small as possible

(b) The following condition is satisfied:
The manifold \( M \overset{\text{def}}{=} f(\chi) \) approximately contains all the sample points:

\[
\{ t_n \}_{n=1}^N \subseteq M
\]

This condition can be restated in a different way:

**The reconstruction error of the sample is small.**

The reconstruction error of the sample is defined as

\[
E_d(\{ t_n \}_{n=1}^N) \overset{\text{def}}{=} \sum_{n=1}^N d(t_n, t_n^*)
\]

where \( t_n^* = f(F(t_n)) \) is the reconstructed vector for point \( t_n \), and \( d \) is a suitable distance in the space \( T \) (e.g. the Euclidean distance in \( \mathbb{R}^B \))[75].

Conditions A and B are not equivalent. B implies A, but not vice versa. This is because, for a given manifold \( M = f(P) \subset T \), \( F \circ f \) need not be the identity mapping, as shown in the Fig 3.1. Thus, it is possible that \( \{ t_n \}_{n=1}^N \subseteq M \) i.e.[75] for each \( t_n \), there exists a point \( x_n \in P \), with \( f(x_n) = t_n \), but \( F(t_n) \neq x_n \).

Therefore, \( t_n^* = f(F(t_n)) \neq t_n \) and \( E_d(\{ t_n \}_{n=1}^N) \) may be large[75].

This is not an arbitrary argument: the dimensionality reduction mappings derived from latent variable models are of this kind, due to the existence of a probability distribution on the space \( P \) and of a noise model[75].

In the Fig 3.1, the dimensionality reduction mappings \( F \) (dashed lines) and the reconstruction mapping \( f \) (thin line) need not verify \( F \circ f = \text{identity}[75] \).

### 3.2.1 The Curse of the Dimensionality

The term curse of the dimensionality refers to the fact, in the absence of simplifying assumptions, the sample size needed to estimate a function of several variables to a given degree of accuracy (i.e. to get a reasonably low variance estimate) grows exponentially with the number of variables. High dimensional spaces are generally sparse. Sometimes in multivariate density estimation, as regions of relatively
very low density can contain a considerable part of the distribution, whereas regions of apparently high density can be completely devoid of observations in a sample of moderate size. For example, for a one-dimensional standard normal $N(0,1)$, 70% of the mass is at points contained in a sphere of radius one standard deviation (i.e. the [-1,1]); for a 10-dimensional $N(0,\mathbf{I})$, that same (hyper)sphere contains only 0.02% of the mass and one has to take a radius of more than 3 standard deviations to contain 70%. Generally in high dimensional distributions the tails are much more important than in one-dimensional ones[75].
Problems with Curse of Dimensionality

(1) Most density estimation methods are based on some local average of the neighbouring observations. In order to find enough neighbours in high-dimensional spaces, the neighbourhood has to reach out farther and the locality is lost.

(2) If there are linear correlations in the data (a very likely situation in high dimensions), the optimal mean integrated squared error when estimating the data density will be very large even if the sample size is arbitrarily large.

It can be said that the curse of dimensionality is governed not by the dimensionality $D$ of the data space $T$, but by the intrinsic dimensionality $L$ of the data.

All dimensionality reduction methods are affected to some extent by the curse of dimensionality (through the number of parameters that need to be estimated)[75].

3.2.2 Intrinsic Dimension of a Sample

Let a phenomenon be governed by $L$ independent variables. In practice, this phenomenon, may have more degrees of freedom due to the influence of a variety of uncontrolled factors: noise, imperfection of the measurement system, addition of irrelevant variables, etc. It can be assumed that this influence is not too strong to completely mask the original structure and it can be filtered out and the original variables or equivalent set of them can be recovered. The intrinsic dimension of a phenomenon is the number of independent variables which explain satisfactorily that phenomenon. The intrinsic dimension would be the dimension $L$ of the manifold that approximately embeds the sample data in a $D$-dimensional Euclidean space ($D > L$). Whenever the intrinsic dimensionality of a data set is smaller than the actual one, dimensionality reduction can bring an improved understanding of the data[49][75][65].

Thus, determination of the intrinsic dimensionality of a process of a given sample, is the central problem in dimensionality reduction, because its knowledge eliminates the possibilities of over or underfitting. A trial and error process is good in order to obtain a satisfactory value for it. In some practical applications, domain information may give insight into the intrinsic dimensionality. For probabilistic methods, such as latent variable models, a Bayesian approach can help to determine the intrinsic dimensionality [2][65][75].
3.3 DIMENSIONALITY REDUCTION METHODS

The methods for reducing the dimensionality of the feature space can be grouped into two categories[2] -

(1) Feature Selection Methods.

(2) Feature Extraction Methods.

The feature selection methods (also known as sub-setting methods), reduce dimensionality by selecting a subset of the original feature set.

The feature extraction methods (also known as transformation methods) reduce the dimensionality by projecting the original $D$-dimensional feature space on a $d$-dimensional subspace (where $d < D$) through a transformation. Each feature in the reduced set is a part of the original feature set when the feature selection methods are used for dimensionality reduction. When feature extraction methods are used, each feature in the reduced feature set is a combination of all the features in the original feature set. The feature selection methods reduce the computational complexity by not computing those features, which are not in the reduced feature set. But in feature extraction methods, all the $D$ features are computed before the dimensionality reduction is performed through transformation[2][54][65].

The efficiency of a feature selection methods depends on how good the figure of merit is in reflecting the effectiveness of individual features. The figure of merit reflects the goodness of an individual feature in the recognition task. The selection of features is done by devising a figure of merit. Once the figure of merit is selected, it is used to rank the $D$ features from the original set and order them. The top $d$, where $d < D$, features are selected from the rank-ordered list. Hence by this approach, only the subset of $d$ individually best features are found. But generally, this subset is not the same as the best subset of $d$ features. In order to find the best subset, one has to inspect all possible subsets, which is computationally very expensive. Some of the figures of merits used in feature selection methods are - F-ratio ( ratio of between-class and within-class variances ), Mahalanobis distance, Bhattacharya distance, Matusita distance, Patrick-Fisher distance, divergence measure, mutual information measure, entropy measure and recognition rate of training data[2]. Transformations in feature extraction methods can be linear or non-linear [26][67][73].
3.4 F-RATIO

F-ratio is defined as the ratio of the between-class variance and the within-class variance. In the context of feature selection for pattern classification, it tries to select the feature which maximizes the separation between different classes and minimizes the scatter within the classes. Using F-ratio as the figure of merit in dimensionality reduction needs the following assumptions[2] -

(a) The feature vectors within each class must have Gaussian distribution.
(b) The features must not be statistically correlated.
(c) The variances within each class must be equal.

But the variances within class are generally not equal, so the pooled within class variance is used to define the F-ratio.

Let \( N \) = number of training patterns in each of the \( K \) classes. It is assumed that \( N \) is the same for \( K \) classes. The F-ratio for the \( i \)-th feature is defined as[2]

\[
F_i = \frac{B_i}{W_i}
\]

where \( B_i \) is the between-class variance and \( W_i \) is the pooled within-class variance. These variances are given by[2] -

\[
B_i = \frac{1}{K} \sum_{k=1}^{K} (\mu_{ik} - \mu_i)^2
\]

\[
W_i = \frac{1}{K} \sum_{k=1}^{K} W_{ik}
\]

where \( \mu_{ik} \) and \( W_{ik} \) are the mean and variance, respectively, of the \( i \)-th feature for the \( k \)-th class, and \( \mu_i \) is the overall mean of the \( i \)-th feature. These are given by[2] -

\[
\mu_{ik} = \frac{1}{N} \sum_{n=1}^{N} x_{ikn}
\]

\[
W_{ik} = \frac{1}{N} \sum_{n=1}^{N} (x_{kn} - \mu_{ik})^2
\]

\[
\mu_i = \frac{1}{K} \sum_{k=1}^{K} \mu_{ik}
\]

where \( x_{ikn} \) is the \( i \)-th feature of the \( n \)-th training pattern from the \( k \)-th class.
3.5 RECOGNITION RATE ON TRAINING DATA

The figure of merit ‘recognition rate on training data’ can be judged by its contribution to the recognition performance. Each feature is used individually to recognize all the patterns in the training set using the pattern recognizer and the recognition rate can be computed for each of the \( D \) features in the original space. The \( D \) vectors are ranked ordered using this recognition rate as the figure of merit, and top \( d (\leq D) \) features are selected\[2\].

This method has an advantage as it makes less assumptions than the \( F \)-ratio based method. It makes the assumption that the features are not statistically correlated, but the other assumptions, that the within class distribution are Gaussian and within-class variances are equal, are not required\[2\].

3.6 LINEAR DISCRIMINANT ANALYSIS

In this method the dimensionality reduction is done by projecting the original \( D \)-dimensional space on a \( d \)-dimensional subspace, where \( d < D \) and finding a linear transformation that defines this sub-space using the LDA. It is assumed that the within class distributions are Gaussian and the within-class covariance matrices are equal\[2][39][78]\.

In LDA, the linear transformation is defined in terms of a rank-ordered set of linearly independent vectors, \( \mathbf{u}_i \), \( i = 1,2, ..., d \). The first of these vectors, \( \mathbf{u}_1 \), is the direction in the original \( D \) dimensional feature space which, when the training vectors are projected onto it, produces the maximum value of the \( F \)-ratio. The second vector \( \mathbf{u}_2 \) is chosen such that it is linearly independent to \( \mathbf{u}_1 \) and produces the next largest \( F \)-ratio. This process is repeated until all the \( d \) linearly independent vector, \( \mathbf{u}_i \), \( i = 1,2, ..., d \) are found\[39][49\]. These vectors can be computed as the \( d \) eigenvectors corresponding to \( d \) largest eigenvalues of the matrix \( \mathbf{W}^{-1} \mathbf{B} \), where \( \mathbf{B} \) is the between-class covariance matrix and \( \mathbf{W} \) is the pooled within-class covariance matrix. These matrices are symmetric and can be computed from the training data as\[2][78\] -

\[
\mathbf{B} = \frac{1}{K} \sum_{k=1}^{K} \left( \mathbf{\mu}_k - \mu \right) \left( \mathbf{\mu}_k - \mu \right)^\top
\]

\[
\mathbf{W} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{W}_k
\]
where $\mu_k$ and $W_k$ are the mean vector and covariance matrix of the $k^{th}$ class, respectively, and $\mu$ is the overall mean. These are given by[2][78] -

$$
\mu_k = \frac{1}{N} \sum_{n=1}^{N} x_{kn}
$$

$$
W_k = \frac{1}{N} \sum_{n=1}^{N} (x_{kn} - \mu) (x_{kn} - \mu_k)^T
$$

$$
\mu = \frac{1}{K} \sum_{k=1}^{K} \mu_k
$$

where $x_{kn}$ is the $n^{th}$ training pattern from the $k^{th}$ class. The number of training patterns in each of the $K$ classes is assumed to be the same ($=N$) for simplifying the presentation.

In linear discriminant analysis, the linear transformation is given by the matrix $U^T$, where $U$ is a $D \times d$ matrix whose columns are the eigenvectors corresponding to the $d$ largest eigenvalues of the matrix $W^{-1}B$. The matrix $U$ is given by[2][78]

$$
U = CL^{-1/2}V
$$

where $C$ is an unitary matrix diagonalizing the within class matrix $W$ to a diagonal matrix $L$. It is $CWC = L$, and $V$ is an unitary matrix whose columns are chosen to be the eigenvectors corresponding to the $d$ largest eigenvalues of the symmetric matrix $S = L^{-1/2}BCL^{-1/2}$. Hence when $D = d$, no dimensionality reduction is performed. the linear transformation computed through linear discriminant analysis is equivalent to a rotation, followed by scaling and then followed by another rotation. Mahalanobis distance remain invariant under this transformation but the Euclidean distance does not remain invariant under this transformation[2].

3.7 PRINCIPAL COMPONENT ANALYSIS

Dimensionality reduction is achieved by projecting the original $D$-dimensional feature space on a $d$-dimensional subspace and finding the orientation of the subspace which best preserves the information available in the original space. The input pattern in
the original $D$-dimensional feature space is transformed to the Karhunen-Loeve (KL) coordinate system and the dimensionality is reduced by representing the pattern by $d$ coordinates in the KL coordinate system. The KL coordinate system represents optimally a set of $D$-dimensional patterns by another set of vectors of a lower dimensionality[39][53][75].

Let the $D$-dimensional patterns are represented by $x_{kn}$, $k=1,2,\ldots,K$, $n=1,2,\ldots,N$ in the training set. The main aim of the PCA is to approximate each $D$-dimensional vector $x_{kn}$ by a $d$-dimensional vector $y_{kn}$ (where $d < D$) such that the mean square error[2]

$$E = \frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} (x_{kn} - Uy_{kn})^t (x_{kn} - Uy_{kn})$$

is minimum[2][60]. Here, $U$ is a $d \times d$ matrix. Here $E$ is minimum if the columns of matrix $U$ are chosen to be the eigenvectors corresponding to the $d$ largest eigenvalues of the total covariance matrix $T$. This matrix is defined as[2][65][75]

$$T = \frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} (x_{kn} - \mu)(x_{kn} - \mu)^t$$

where $\mu$ is the overall mean given by

$$\mu = \frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} x_{kn}$$

Hence the linear transformation is given by the matrix $U^t$, where $U$ is a $D \times d$ unitary matrix whose columns are the eigenvectors corresponding to the $d$ largest eigenvalues of the total covariance matrix $T$. When no dimensionality reduction is performed, (when $d = D$), this transformation amounts to a rotation in the feature space. Thus the Mahalanobis distance and the Euclidean distance remain invariant under this transformation[2].

This method neither maximizes the between class separation nor minimizes the within class scatter. It only minimizes the mean square error in approximating the set of training vectors by another set of vectors of a lower dimensionality[2].

3.7.1 Geometrical Properties of Principal Components

Geometrical properties of principal components can be described by some two dimensional figures. Let there be a sample of observations on two standardized variables $x_1$ and $x_2$. The $x_1$ and $x_2$ can be used as coordinate axes and a plot of the
standardized scores can be shown as in Fig 3.2. From the shape of the scatterplot, it is obvious that there is a substantial correlation between $x_1$ and $x_2$. Since there are two variables, two principal components are required to completely account for the variation in the two variables[69][73].

The first principal component is a new coordinate axis in the variable space which is oriented in a direction that maximizes the variation of the projections of the points on the new coordinate axis. The projection of a point on a coordinate axis is the numerical value on the coordinate axis at which a line from the point drawn perpendicular to the axis intersects the axis. The distribution of the points makes an elliptical shape and the first principal component is the principal axis of this ellipse. For two standardized variables, the first principal component always forms a 45° angle with $x_1$ and thus with $x_2$, irrespective of the size of the correlation as long as the correlation is not zero. The latent vector associated the largest root of the two by two correlation matrix is always[69]

$$a_1 = \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix}$$

![Fig 3.2 Scatter plot of two standardized variables (after[69])](image-url)
There are as many direction cosines for a particular principal component as there are variables (the number of elements in each latent vector). Thus, the elements of the latent vector are the direction cosines of the first principal component with the $x_1$ and $x_2$ axes, respectively. The largest principal component makes an angle of 45° with both $x_1$ and $x_2$ and the cosine of 45° is 0.71. The length of each latent vector is 1 and if it this unit length vector is perpendicularly projected on both $x_1$ and $x_2$, the respective projections are each equal to 0.71. Hence, the projections of the unit length latent vector variable representing a particular principal component, on each of the variables coordinate axes are equal to the direction cosines.

For more than two variables, the elements of the latent vector for the largest principal component usually differ from one another. The large correlation matrices rarely have all correlations equal to one another.

If the correlation between the two variables is high, then most of the information contained in the two dimensional distribution of observations can be represented by their projection onto the single largest principal component. In Fig 3.3, eight observations represent an almost perfect correlation between $x_1$ and $x_2$. The dotted lines represent the perpendicular projections of these points. These projections are the principal component scores of the linear transformation $y_1 = 0.71x_1 + 0.71x_2$. It is

![Fig 3.3 1-D Representation by Largest Principal Components of 2-D Data (after[69])](image-url)
interesting to note that the principal scores for the largest principal component contain as much information regard to the scatter of the observation as the two dimensional scatter in the space of the original two variables, $x_1$ and $x_2$. The reason for this is - the observations fall upon a straight line (the largest principal component vector). Thus the information concerning the scatter in a two dimensional space can be captured by projecting the points onto a one-dimensional subspace (the largest principal component)[69][73].

The first principal component also minimizes the sum of the squared distances of the observations from their perpendicular projections onto the largest principal component. Because of this, the principal component analysis is sometimes called as ‘the line of closest fit’. The second principal component is orthogonal to the first principal component[69][73][74].

When two variables have a zero correlation, then the distribution of observations in the variable space has circular shape as shown in Fig 3.4. Here, there is no coordinate axis that maximizes the variance of the projections upon it. There is no unique direction in which the coordinate axis can be placed to maximize variation. One

![Fig 3.4 Distribution of Two Uncorrelated Variables (after[69])](image)
direction is as good as another. Thus, the direction cosines associated with the largest latent root are not unique. There are an infinite number of directions that are represented by the latent vector for the largest principal component. Irrespective of the coordinate axis, the variance of the projections of the points on the coordinate axis is always the same because the spread of the points on each possible coordinate axis is equal to the diameter of the circle representing the distribution of points.
Regression analysis for the two variable cases also results in a line of closest fit. The regression line is obtained by minimizing the sum of squared distances of the points from the line in the direction of the dependent variable [Fig 3.5]. These distances are not perpendicular to the line as they are in the principal component analysis[69][73].

When the variances of one variable is a large multiple of the other variables, then the direction cosine for that variable will be close to 1 and the direction cosines associated with the remaining variables will be close to zero. Hence, the principal component is oriented in the direction of the variable with the largest variance. As shown in Fig 3.6, this is the direction that maximizes the variation of the largest principal component. The variance of \( x_1 \) is high as compared to \( x_2 \). The largest component has a large direction cosine for \( x_1 \) and a small one for \( x_2 \) [69].

### 3.8 FACTOR ANALYSIS

Factor analysis consists of procedures for analysing the relations among a set of random variables observed, counted or measured for each individual of a group. An individual may be a person, some other organism, an object, or in general any entity. A group consists of one class of such entities[73][74].

The purpose of a factor analysis is to account for the intercorrelations among \( n \) variables, by postulating a set of **common factors**, considerably fewer in number than the number, \( n \), of these variables. The factors are random variables that cannot be observed, counted or measured directly, but which are presumed to exist in the population and hence they are in the experimental sample. They, as well as the manifest variables, are presumed to vary from individual to individual. Factors are also sometimes called as **latent variables**. The factors are actually hypothetical or explanatory constructs. In factor analysis, interest centers mainly on the common factors, which are interpreted with reference to the observed or counted or measured variables. The unique factors in each variable is merely whatever part of that variable is uncorrelated with all the other variables in the total collection of observed variables, including its error of measurement[73][75].

### 3.9 PRINCIPAL COMPONENT v/s FACTOR ANALYSIS

Principal Component Analysis and Factor Analysis, both can be used for dimensionality reduction but there are some important differences between the two techniques. The principal component analysis decomposes the total variance and in the
case of standardized variables, it produces a decomposition of $R$ (correlation matrix). The $R$ can be decomposed into the product of three matrices $A \Lambda A'$, where $A$ is the matrix of the latent vectors $R$ arrayed as columns, $A'$ is its transpose (latent vectors arrayed as rows) and $\Lambda$ is a diagonal matrix of the associated latent roots of $R$. Because $\Lambda$ is a diagonal matrix, such that

$$\sum_{i=1}^{p} \lambda_i a_i a_i' = \mathbf{1}_i$$

so that in terms of $\mathbf{1}_i$, $R$ can be decomposed as

$$\sum_{i=1}^{p} \mathbf{1}_i \mathbf{1}_i'$$

or $LL'$, where $L$ is the principal component loading matrix with principal components as columns and $L'$ is its transpose. The $R$ can be approximated to any degree by retaining $k$ principal components and discarding the remaining $p - k$. In this way, $R$ can be approximated as

$$L \times L'$$

$$(p \times k) \times (k \times p)$$

Factor analysis finds the decomposition of the reduced correlation matrix $R-U$, where $U$ is the diagonal matrix of the unique variances associated with the variables. Unique variances are that part of each variable’s variance that has nothing in common with the remaining $p - 1$ variables. If the unique variances are subtracted from 1, the elements in the principal diagonal of $R$, we get the communalities in the principal diagonal of the reduced correlation matrix. The communality for each variable is the portion of the total variance that is shared with the remaining $p - 1$ variables. Let $R_c$ be the reduced correlation matrix with communalities in the principal diagonal instead of ones. Factor analysis seeks a decomposition of $R_c$ in terms of a $p \times k$ factor loading matrix $L_f$ with the smallest $k$ such that $R_c = L_f L_f'$. Usually, $k$ is much smaller than $p$. This contrasts with principal component analysis where $L$ must be a $p \times p$ matrix in order for the principal components to fit $R$ perfectly. In factor analysis, it is theoretically possible to find a $p \times k$ factor loading matrix $L_i$ where $k$ is much smaller than $p$ that fits
The principal components can be expressed as linear functions of the variables or the variables can be expressed as linear functions of the principal components. But this is not the case with factor analysis because factor analysis concentrates on defining the variables as a linear combination of common factors and unique factors. Contrary to principal analysis, the factor analysis model does not provide a unique transformation from variables to factors[69][73].

Correlation matrix is decomposed without regard to an underlying model in principal component analysis. It does not distinguish between common variance and unique variance as factor analysis does. Where as factor analysis, has an underlying model that rests on a number of assumptions. The key assumption is that the $i^{th}$ variable in the variable set $x_i$, can be expressed as a linear combination of hypothetical unobservable common factors plus a factor unique to that variable. The emphasis is on explaining $x_i$ as a linear function of unobservable common factors while the emphasis in principal component analysis is expressing the principal component as a linear function of the $x_i$. This model assumes that the standardized variable $x_i$, can be expressed as

$$
\sum_{j=1}^{k} l_{ij} f_j + u_i
$$

where $l_{ij}$ is the loading of the $i^{th}$ variable on the $j^{th}$ factor, $f_j$ is the factor score for the $j^{th}$ factor, and $u_i$ is the unique factor score for the $i^{th}$ variable.

In factor analysis, the loadings $l_{ij}$ are estimated which together form the factor loading matrix $L_f$ which has $p$ rows corresponding to variables and $k$ columns corresponding to factors where $k$ is considerably smaller than $p$. To estimate $L_f$, a number of assumptions are being made. As described in [69], these assumptions are -

1. the mean and variance of $f_j$ equal to zero and one, respectively for all $j$,
2. the correlations among the common factors are zero,
3. the correlations between the common factors and the unique factors are zero,
4. the correlations between the $p$ unique factors are all zero.
When there is no underlying model of the data in mind, then principal components can be useful in exploring the structure of a data set. It can be used directly to construct linear composites in order to both simplify the description of a data set and to be used as independent and dependent variables in further analysis [69][74].

If the variables contain a substantial amount of measurement error and an underlying model can be postulated, then factor analysis can be advantageous over principal component analysis. The common factors are uncontaminated by measurement error because measurement error is part of the unique variance which is uncorrelated with the common variance. Since, in this case, the principal components will be linear composites of unreliable variables, the principal components will contain measurement error [69][73][74].

3.9.1 Shortcomings of PCA

Current Gaussian-mixture-models (GMM) based classifiers are the most popular classifiers in the pattern recognition literature. In order to train Gaussian-mixture-model parameters, we need probability density function of a feature vector so that the model parameters can be computed by maximizing the likelihood of the training vectors.

In PCA, dimensionality of the feature vectors is minimized by minimizing the reconstruction error between the original vectors and the reconstructed vectors. It does not provide a probabilistic framework for computing the maximum likelihood estimates of the parameters.

In order to apply PCA with GMM classifiers, we need a probabilistic framework to compute the maximum likelihood estimates of the parameters. Unfortunately, PCA is not amicable for this purpose. Recently, factor analysis (FA) has been developed to compute the maximum likelihood estimates of the parameters.

3.10 SUMMARY

Factor analysis provides a unified probabilistic model for computing the maximum likelihood estimates of the parameters. Using expectation-maximization algorithm, factor analysis can be useful in overcoming the shortcomings of PCA. Factor analysis model, based upon Expectation-maximization algorithm, is described in the next chapter.
The factor analysis describes the covariance relationships among many variables in terms of a few underlying, but unobservable, random quantities called factors. Factor analysis believes that the variables can be grouped by their correlations.

It may be assumed that variables within a particular group are highly correlated among themselves, but they have relatively small correlations with variables in a different group. Then it can be said that each group of variables represents a single underlying construct (or factor) that is responsible for the observed correlations[73]. Factor analysis can be viewed as an attempt to approximate the covariance matrix $\Sigma$.

**4.1 ORTHOGONAL FACTOR MODEL**

Let

$X$ : be an observable random vector with $p$ components

$\mu$ : mean of $X$

$\Sigma$ : covariance of $X$

$F_1, F_2 \ldots F_m$ : unobservable random variables called *common factors*

$\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_p$ : additional sources of variations called *errors or specific factors*

The factor model postulates that $X$ is linearly dependent upon common factors and errors, and the factor analysis model can be described as[73] –
\[
X_1 - \mu_1 = l_{11}F_1 + l_{12}F_2 + \ldots + l_{1m}F_m + \varepsilon_1
\]
\[
X_2 - \mu_2 = l_{21}F_1 + l_{22}F_2 + \ldots + l_{2m}F_m + \varepsilon_2
\]
\[
X_1 - \mu_1 = l_{31}F_1 + l_{32}F_2 + \ldots + l_{3m}F_m + \varepsilon_3
\]
\[
X_p - \mu_p = l_{p1}F_1 + l_{p2}F_2 + \ldots + l_{pm}F_m + \varepsilon_p
\]
or
\[
X - \mu = L F + \varepsilon
\]
\[
(\text{p} \times 1) \quad (\text{p} \times \text{m}) \quad (\text{m} \times 1) \quad (\text{p} \times 1)
\]

The coefficient \( l_{ij} \) is called the \textit{loading} of the \textit{i}th variable on the \textit{j}th factor, so that the matrix \( L \) is the matrix of factor loadings. The \( p \) deviations \( X_1 - \mu_1, X_2 - \mu_2, X_3 - \mu_3, \ldots, X_p - \mu_p \) are expressed in terms of \( p + m \) random variables \( F_1, F_2, F_3, \ldots, F_m, \varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots, \varepsilon_p \) which are \textit{unobservable}[73].

The unobservable random vectors \( F \) and \( \varepsilon \) satisfy the following conditions -
\[
E(F) = 0, \quad \text{Cov}(F) = E[FF'] = I
\]
\[
(\text{m} \times 1) \quad (\text{m} \times \text{m})
\]
\[
E(\varepsilon) = 0, \quad \text{Cov}(\varepsilon) = E[\varepsilon\varepsilon'] = \Psi = \begin{bmatrix}
\psi_1 & 0 & 0 & \ldots & 0 \\
0 & \psi_2 & 0 & \ldots & 0 \\
0 & 0 & \psi_3 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & \ldots & \psi_p
\end{bmatrix}
\]
\[
(\text{p} \times 1) \quad (\text{p} \times \text{p})
\]

\[\text{......................... (4-3)}\]
and that $F$ and $\varepsilon$ are independent, so
\[
\text{Cov}(\varepsilon, F) = E(\varepsilon F') = 0
\]
\[(p \times m)\]
where $\Psi$ is a diagonal matrix.

4.1.1 Covariance Structure for the Orthogonal Factor Model

The orthogonal factor model implies a covariance structure for $X$ [73].

\[
(X - \mu)(X - \mu)' = (LF + \varepsilon)(LF + \varepsilon)
\]
\[
= (LF + \varepsilon)(LF)' + \varepsilon LF' + LF\varepsilon' + \varepsilon\varepsilon'
\]
so that
\[
\Sigma = \text{Cov}(X) = E(X - \mu)(X - \mu)'
\]
\[
= LE(FF)'L' + E(\varepsilon F')L' + LE(Fe') + E(\varepsilon\varepsilon')
\]
\[
= LL' + \Psi
\]

Also,
\[
(X - \mu)F' = (LF + \varepsilon)F'
\]
\[
= LFF' + \varepsilon F'
\]
so
\[
\text{Cov}(X,F) = E(X - \mu)F'
\]
\[
= LE(FF') + E(\varepsilon F')
\]
\[
= L
\]

The model $X - \mu = LF + \varepsilon$ is linear in the common factors. The portion of the variance of the $i^{th}$ variable contributed by the $m$ common factors is called the $i^{th}$ communality. That portion of $\text{Var}(X_i) = \sigma_i$ due to the specific factor is often called the specific variance [73].
Let the $i^{th}$ communality is denoted by $h_i^2$,

$$\sigma_{ii} = l_{i1}^2 + l_{i2}^2 + \ldots + l_{im}^2 + \Psi_{ii} \quad \text{.................. (4-5)}$$

$$\text{Var}(X_i) = \text{communality} + \text{specific variance}$$

or

$$h_i^2 = l_{i1}^2 + l_{i2}^2 + \ldots + l_{im}^2 \quad \text{.................. (4-6)}$$

and

$$\sigma_{ii} = h_i^2 + \Psi_i, \quad i = 1, 2, 3, \ldots, p \quad \text{.................. (4-7)}$$

The $i^{th}$ communality is the sum of squares of the loadings of the $i^{th}$ variable on the $m$ common factors.

The factor model assumes that the $p + p(p - 1)/2 = p(p + 1)/2$ variances and covariances for $X$ can be reproduced from the $pm$ factor loadings $l_{ij}$ and $p$ specific variances $\Psi_i$. When $m = p$, any covariance matrix $\Sigma$ can be reproduced exactly as $LL'$, so $\Psi$ can be a zero matrix. However, it is when $m$ is small relative to $p$ that factor analysis is most useful. In this case, the factor analysis model provides a 'simple' explanation of the covariation in $X$ with less parameters than the $p(p + 1)/2$ parameters in $\Sigma$ [73].

When $m > 1$, there is always some inherent ambiguity associated with the factor model. Let $T$ be any $m \times m$ orthogonal matrix, so that $TT' = T'T = I$.

Then[73]

$$X - \mu = LF + \epsilon \quad \text{.................. (4-8)}$$

$$= LTT'F + \epsilon$$

$$= L^*F^* + \epsilon$$

where

$$L^* = LT \quad \text{and} \quad F^* = T'F$$

Since

$$E(F^*) = T'^e(F) = 0$$

and
Cov (F') = T'Cov(F)T = T'T = I 

(m x m)

which is impossible on the basis of observations on X, to distinguish the loadings L from the loadings L'. That is the factors F and F' = T'F have the same statistical properties, and even though the loadings L' are, in general, different from the loadings L, they both generate the same covariance matrix Σ [73].

Σ = LL' + Ψ 

.............................. (4-9)

= LTT'L' + Ψ 

= (L') (L')' + Ψ 

Because of this ambiguity, there is a need for "factor rotation", because orthogonal matrices correspond to rotations (and reflections) of the coordinate system for X [73].

The factor model allows to uniquely estimate L and Ψ. The loading matrix is then rotated (multiplied by an orthogonal matrix), where the rotation is determined by some criterion[73].

4.2 METHODS OF ESTIMATION

If Σ appears to deviate significantly from a diagonal matrix, then factor model can be entertained, and the initial problem is one of estimating the factor loadings l_{ij} and specific variances Ψ_i.

There are two most commonly used methods for parameter estimation[73]:-

1. Principal Factor Method.


The solution from either method can be rotated in order to simplify the interpretation of factors.

4.2.1 Principal Factor Method

Let Σ have eigenvalue-eigenvector pairs (λ_i, e_i) with λ_1 ≥ λ_2 ≥ λ_3 ...... ≥ λ_p ≥ 0. Then[73]
\[ \Sigma = \lambda_1 e_1 e_1' + \lambda_2 e_2 e_2' + \cdots + \lambda_p e_p e_p' \]

\[ = \left[ \sqrt{\lambda_1} e_1 \mid \sqrt{\lambda_2} e_2 \mid \cdots \mid \sqrt{\lambda_p} e_p \right] \left[ \begin{array}{c} \sqrt{\lambda_1} e_1' \\ \sqrt{\lambda_2} e_2' \\ \sqrt{\lambda_3} e_3' \\ \vdots \\ \sqrt{\lambda_p} e_p' \end{array} \right] \]

\[ \text{................. (4-10)} \]

This satisfies the covariance structure for the factor analysis model having as many factors as variables \((m = p)\) and specific variances \(\Psi_i = 0\) for all \(i\). It can be written

\[ \Sigma = L L' + 0 = L L' \text{ ................. (4-11)} \]

\((p \times p) \quad (p \times p) \quad (p \times p) \quad (p \times p)\)

When the last \(p - m\) eigenvalues are small, we can obtain the approximation as

\[ \text{................. (4-12)} \]

\((p \times m) \quad (m \times p)\)

The above approximate representation assumes that the specific factors \(e\) are of minor importance and can also be ignored in the factoring of \(\Sigma\). If specific factors are included in the model, their variances may be taken to be the diagonal elements of \(\Sigma - LL'\), where \(LL'\) is defined in the above equation[73].

When specific factors are included, the approximation becomes[73]

\[ \Sigma = LL' + \Psi \text{ ................. (4-13)} \]
When the above representation is applied to the sample covariance matrix $\Sigma$, it is known as the principal component solution. The factor loadings are the scaled coefficients of the first few sample principal components\[73\].

To apply this approach to a data set $x_1, x_2, \ldots, x_n$, it is generally done to center the observations by subtracting the sample mean. The centered observations are\[73\]-

For the principal component solution, the estimated loadings for a given factor do not change as the number of factors is increased.
4.2.2 The Maximum Likelihood Method

When the common factors $F$ and the specific factors $\varepsilon$ are assumed to be normally distributed, then maximum likelihood estimates of the factor loadings and specific variances can be obtained. When $F_j$ and $\varepsilon_j$ are jointly normal, the observations $X_j - \mu = LF_j + \varepsilon_j$ are then normal and the likelihood is

$$L(\mu, \Sigma) = \left(\frac{2\pi}{np}\right)^{-n/2} |\Sigma|^{-1/2} e^{-\frac{1}{2} \text{tr}[\Sigma^{-1} \left( \sum_{j=1}^{n} (x_j - \bar{x})(x_j - \bar{x})' + n(\bar{x} - \mu)(\bar{x} - \mu)' \right)]}$$

$$= \left(\frac{2\pi}{n-1}\right)^{-p/2} |\Sigma|^{-1/2} e^{-\frac{1}{2} \text{tr}[\Sigma^{-1} \left( \sum_{j=1}^{n} (x_j - \bar{x})(x_j - \bar{x})' \right)]}$$

$$\times \left(\frac{2\pi}{n-1}\right)^{-p/2} |\Sigma|^{-1/2} e^{-\frac{1}{2} \text{tr}[\Sigma^{-1} (\bar{x} - \mu)(\bar{x} - \mu)']}$$

which depends on $L$ and $\Psi$ through $\Sigma = LL' + \Psi$. In this model, $L$ can be well defined by imposing the computationally convenient uniqueness condition

$$L ' \Psi^{-1} L = \Delta$$

a diagonal matrix

$$\text{............... (4-17)}$$

The maximum likelihood estimates of $L$ and $\Psi$ must be obtained by the numerical maximization of the above equation.

Factor Rotation

All factor loadings obtained from the initial loadings by an orthogonal transformation have the ability to reproduce the covariance matrix. An orthogonal transformation corresponds to a rigid rotation of the coordinate axes. An orthogonal transformation of the factor loadings is called factor rotation [73].

Let

$$L^*: \text{ be a } p \times m \text{ matrix of estimated factor loadings obtained by any method.}$$

then

$$L^* = LT, \text{ where } TT' = T'T = I$$

$$\text{............... (4-18)}$$

is a $p \times m$ matrix of “rotated” loadings.
Since the original loadings may not be readily interpretable, it is usual practice to rotate them until a “simpler structure” is achieved.

When \( m = 2 \), or the common factors are considered two at a time, the transformation to a simple structure can frequently be determined graphically. The uncorrelated common factors are regarded as unit vectors along perpendicular coordinate axes. A plot of the pairs of factor loadings \((l_{1i}, l_{2i})\) yields \( p \) points, each point corresponding to a variable. The coordinate axes can then be visually rotated through an angle - say \( \varphi \), and the new rotated loadings are determined from the relationships

\[
L' = L \cdot T
\]

\[
T = \begin{bmatrix}
\cos \varphi & \sin \varphi \\
-\sin \varphi & \cos \varphi
\end{bmatrix}
\]
clockwise rotation

\[
T = \begin{bmatrix}
\cos \varphi & -\sin \varphi \\
\sin \varphi & \cos \varphi
\end{bmatrix}
\]
anti-clockwise rotation

For \( m > 2 \), orientations are not easily visualized, and the magnitudes of the rotated loadings must be inspected to find a meaningful interpretation of the original data. Rotation of factor loadings is found useful for loadings obtained by maximum likelihood, since the initial values are constrained to satisfy the uniqueness condition that \( L'\Psi^{-1}L \) be a diagonal matrix[73].

Orthogonal rotations are appropriate for a factor model in which the common factors are assumed to be independent. Oblique rotations are non-orthogonal rotations but not frequently used in factor analysis. If we regard the \( m \) common factors as coordinate axes, the point with the \( m \) coordinates represents the position of the \( i \)th variable in the factor space. Assuming that the variables are grouped into non-overlapping clusters, an orthogonal rotation to a simple structure corresponds to a rigid rotation of the
coordinate axes such that the axes, after rotation, pass as closely to the clusters as possible. An oblique rotation is a simple structure that corresponds to a nonrigid rotation of the coordinate system so that the rotated axes (no longer perpendicular) pass nearly through the clusters. An oblique rotation seeks to express each variable in terms of a minimum number of factors - preferably, a single factor[73].

**Factor Scores**

In factor analysis, interest is usually centered on the parameters in the factor model. Sometimes, the estimated values of the common factors, called factor scores, may subsequent analysis[73].

Factor scores are not estimates of unknown parameters in the usual sense. They are rather estimates of the values for the unobserved random vectors \( F_j, j = 1,2,3, \ldots , n \).

There are two approaches to find the factor scores[73] --

1. Weighted Least Squares Method
2. Regression Method

Both the factors score approaches have two elements in common[73] -

(i) They treat the estimated factor loadings \( l_{ij} \) and the specific variances \( \Psi_i \), as if they were true values.

(ii) They involve linear transformation of data.

### 4.3 EXPECTATION - MAXIMIZATION ALGORITHM

In signal processing applications it is quite common to estimate the parameters of a probability distribution function. In many parameter estimation problems, the situation is quite complicated as direct access to the data necessary to estimate the parameters is impossible, or some of the data are missing. Such difficulties arise when an outcome is a result of an accumulation of simpler outcomes, or when outcomes are clumped together. There may also be data dropouts or clustering in such a way that the number of underlying data points is unknown. The EM (expectation maximization) algorithm is ideally suited to problems of this sort, in that it produces maximum-likelihood (ML) estimates of parameters when there is a many-to-one mapping from an underlying distribution to the distribution governing the observation. The EM algorithm
consists of two steps: an expectation step, followed by a maximization step. The expectation step is with respect to the unknown underlying variables, using the current estimate of the parameters and conditioned upon the observations. The maximization step then provides a new estimate of the parameters. These two steps are iterated until convergence, as shown in the Fig 4.1.

Dempster, Laird and Rubin [81] brought the ideas of EM algorithm together and proved the convergence and coined the term “EM algorithm”. Since then, a large number of papers employing the EM algorithm in many areas have been published. EM algorithm has also been widely used in econometrics, clinical and sociological studies, genetics, training of hidden Markov models in speech recognition, pattern recognition and neural network training, image modeling, etc. [81][82][89]

Choose an initial parameter $\Theta_0$

Set $t=0$

E-Step
Estimate Unobserved Data using $\Theta_t$

M-Step
Compute Maximum Likelihood Estimate of Parameter $\Theta_{t+1}$ using estimated data

$t = t + 1$
Converged?

Y

Figure 4.1 An overview of the EM algorithm (after[89])
Dempster, Laird and Rubin [81] have proved certain properties of EM algorithm in particular that at each iteration the log-likelihood of the observed data is guaranteed to be non-decreasing. That is, if \( L(\Theta) \) is the likelihood of the observed data given parameter values \( \Theta \), and \( \Theta_t, \Theta_{t+1} \) are the parameter values at the \( t \)’th and \( t+1 \)’th iterations respectively, then \( L(\Theta_{t+1}) \geq L(\Theta_t) \). They also defined Generalised EM (GEM) algorithms, which include EM as a special case, and can be more computationally efficient, while still guaranteeing that \( L(\Theta_{t+1}) \geq L(\Theta_t) \). [81][82][89]

4.3.1 Maximum - likelihood Estimation

In general we have

(a) a sample \( X = \{X_1, X_2, \ldots, X_n\} \) where each \( X_i \) is a random variable (a single value, or vector of values).

(b) a vector of parameters \( \Theta \) such that we can define the likelihood of the data \( P(X|\Theta) \). We can also define the log-likelihood \( L(X|\Theta) = \log P(X|\Theta) \). Often the \( X_i \)’s are independently identically distributed so that

\[
L(X|\Theta) = \sum_{i=1}^{n} \log P(X_i|\Theta)
\]

If \( \Omega \) is the parameter space, maximum - likelihood (ML) estimation involves setting the ML estimate \( \Theta_{ML} \) such that

\[
\Theta_{ML} = \arg \max_{\Theta \in \Omega} L(X|\Theta)
\]

4.3.2 The EM Algorithm

The EM algorithm assumes the following problem definition: we have two sample spaces \( P \) and \( K \), such that there is a many-one mapping \( Y = f(X) \) from an observation \( X \) in \( P \) to an observation \( Y \) in \( K \). We define

\[
P(Y) = \{ X : f(X) = Y \}
\]

\( X \) is the complete data, and \( Y \) is the observed data. If the distribution \( f(X|\Theta) \) is well defined then the probability of \( Y \) given \( \Theta \) is

\[
g(Y|\Theta) = \int_{f(Y)} f(X|\Theta) \, dX
\]

EM attempts to solve the following problem: given a sample from \( Y \) is observed, but the corresponding \( X \) are unobserved, or hidden, find the maximum - likelihood
estimate \( \Theta_{\text{ML}} \) which maximizes \( L(\Theta) = \log g(Y|\Theta) \). In general, \( \log f(X|\Theta) \) will have an
easily-defined, analytically solvable maximum, but maximization of \( L(\Theta) \) has no
analytic solution. EM is an iterative optimization algorithm which defines a sequence of
parameter settings through a mapping \( \Theta_t \rightarrow \Theta_{t+1} \) such that \( L(\Theta_{t+1}) \geq L(\Theta_t) \) with equality
holding only at stationary points of \( L(\Theta) \). Thus EM is a hill-climbing algorithm which, at
least under certain conditions, will converge to a stationary point of \( L(\Theta) \). [81][82][89]

The mapping \( \Theta_t \rightarrow \Theta_{t+1} \) is defined in two steps:

1. \textit{The Estimation Step}: Define

   \[
   \tilde{p}(X) = p(X|Y, \Theta_t)
   \]

   We know that \( \tilde{p}(X) = 0 \) outside \( P(Y) \)

   Calculate

   \[
   Q(\Theta', \Theta_t) = E[\log f(X|\Theta') | \tilde{p}(X)]
   = \int \tilde{p}(X) \log f(X|\Theta') \, dX
   \]

2. \textit{The Maximization Step}:

   \[
   \Theta_{t+1} = \arg \max_{\Theta'} Q(\Theta', \Theta_t)
   \]

   The intuition is as follows: if we had the complete data, we would simply estimate \( \Theta' \) to
maximize \( \log f(X|\Theta') \). But with some of the complete data missing we instead maximize
the expectation of \( \log f(X|\Theta') \) given the observed data and the current value of \( \Theta \).

4.3.3 \( L(\Theta) \) is non-decreasing at each iteration

An important property of the EM algorithm is that log-likelihood \( L(\Theta) = \log g(Y|\Theta) \) is non-decreasing at each iteration.

Formally, if we define the EM mapping as \( \Theta_t \rightarrow \Theta_{t+1} \), where

\[
\Theta_{t+1} = \arg \max_{\Theta'} Q(\Theta', \Theta_t)
\]

then

\[
L(\Theta_{t+1}) \geq L(\Theta_t)
\]

The proof of this property depends upon the following two results:

(1) Define \( k(X|Y, \Theta) \) to be the posterior likelihood of the complete data given the data \( Y \)
and the parameters $\Theta$, so that

$$k(X \mid Y, \Theta) = \frac{f(X \mid \Theta)}{g(Y \mid \Theta)}$$

If we define

$$H(\Theta', \Theta) = E \left[ \log k(X \mid Y, \Theta') \mid \bar{p}(X) \right]$$

\begin{equation}
\text{.......................... (4-27)}
\end{equation}

with equality iff $\log k(X \mid Y, \Theta') = \log k(X \mid Y, \Theta)$ almost everywhere.

From eq(4-27),

$$L(\Theta_{t+1}) - L(\Theta_t) = \{Q(\Theta_{t+1}, \Theta_t) - Q(\Theta_t, \Theta_t)\} - \{H(\Theta_{t+1}, \Theta_t) - H(\Theta_t, \Theta_t)\}$$

But

$$\{Q(\Theta_{t+1}, \Theta_t) - Q(\Theta_t, \Theta_t)\} \geq 0 \quad \text{(by the definition of M step)}$$

\begin{equation}
\text{.......................... (4-29)}
\end{equation}

From eq(4-28)

$$\{H(\Theta_{t+1}, \Theta_t) - H(\Theta_t, \Theta_t)\} \leq 0$$

Hence

$$L(\Theta_{t+1}) - L(\Theta_t) \geq 0 \quad \text{.......................... (4-30)}$$

Proof of equation

By the rules of conditional probability

$$\log k(X \mid Y, \Theta') = \log f(X \mid \Theta') - \log g(Y \mid \Theta')$$

\begin{equation}
\text{.......................... (4-31)}
\end{equation}

We can now take expectations with respect to

$$E \left[ \log k(X \mid Y, \Theta') \mid \bar{p}(X) \right] = E \left[ \log f(X \mid \Theta') \mid \bar{p}(X) \right] - E \left[ \log g(Y \mid \Theta') \mid \bar{p}(X) \right]$$

$$= E \left[ \log f(X \mid \Theta') \mid \bar{p}(X) \right] - \log g(Y \mid \Theta')$$

\begin{equation}
\text{.......................... (4-32)}
\end{equation}
Since
\[ E \left[ \log g(Y | \Theta') \right] = \log g(Y | \Theta') \] 
.......................... (4-33)
as long as \( \log g(Y | \Theta) \) does not depend on \( X \). Hence by the definitions of \( H, Q \) and \( L \),

Proof of equation (4-33)
The equation \( H(\Theta | \Theta) - H(\Theta' | \Theta) \) is the Kullback-Liebler distance between \( k(X | Y, \Theta) \) and \( k(X | Y, \Theta') \), which is known to be \( \geq 0 \) with equality only if the two distributions are equal.

Let \( f(X) \) and \( g(X) \) be non-negative and integrable functions, and \( S \) be the region in which \( f(X) > 0 \). The theorem states that if

\[ \int_S (f(X) - g(X)) \, dX \geq 0 \]

then

\[ \int_S f(X) \log \frac{f(X)}{g(X)} \, dX \geq 0 \]

we put \( f(X) = k(X | Y, \Theta) \) and \( g(X) = k(X | Y, \Theta') \), then clearly

\[ \int_S (f(X) - g(X)) \, dX \geq 0 \]
as \( \int_S f(X) \, dX = 1 \) and by the laws of probability \( \int_S g(X) \, dX \leq 1 \). Hence

\[ \int_S f(X) \log \frac{f(X)}{g(X)} \, dX = \int_S k(X | Y, \Theta) \log \frac{k(X | Y, \Theta)}{k(X | Y, \Theta')} \geq 0 \]

But

\[ H(\Theta, \Theta) - H(\Theta', \Theta) = E \left[ \log k(X | Y, \Theta) \right] - E \left[ \log k(X | Y, \Theta') \right] \]

\[ = \int_S k(X | Y, \Theta) \log k(X | Y, \Theta) - \int_S k(X | Y, \Theta) \log k(X | Y, \Theta') \]

\[ = \int_S k(X | Y, \Theta) \log k(X | Y, \Theta) \frac{k(X | Y, \Theta')}{k(X | Y, \Theta')} \]

\[ \geq 0 \]

.......................... (4-36)

.......................... (4-37)
Hence, the Expectation - Maximization (EM) algorithm is a parameter estimation method which falls into the general framework of maximum-likelihood estimation, and is applied in cases where part of the data can be considered to be incomplete, or “hidden.” It is essentially an iterative optimization algorithm which, at least under certain conditions, will converge to parameter values at a local maximum of the likelihood function. [81][82][89]

4.4 SHORTCOMINGS OF GLOBAL DIMENSION REDUCTION METHODS

In principle global dimensionality reduction methods are capable of obtaining low dimensional encodings, but in practice global dimensionality reduction produce encodings with a high error and take long time to train. A global model may be inappropriate for some datasets. In some datasets, the data has a different structure of the data and different (local) dimensionality in different regions of the input space. Global dimensionality reduction can be inefficient in such datasets. Local models can generate more accurate fits for such datasets because an inaccurate fit in one (local) region of the input space does not effect the fit in other regions of the input space for local models.

4.5 SUMMARY

Global models for low dimensional encoding can be inefficient for some datasets with different local structures. Local techniques, using vector quantization, may provide more accurate encodings in these situations. We will study the vector quantization method and its application for local dimensionality reduction in the next chapter, where two methods namely VQPCA (Vector Quantization Principal Component Analysis) and VQ_EM_FA (Vector Quantization Expectation-Maximization based Factor Analysis) are described. In VQPCA technique, VQ is a hard clustering method that uses minimum square error criterion for codebook generation. The VQPCA method cannot be generalized to get a model which maximizes the likelihood of the data. Thus, maximum-expectation algorithm based factor analysis model can be used to maximize the likelihood of the data.
The goal of dimensionality algorithms is to obtain a compact and accurate representation of multi-variate data that reduces or eliminates statistically redundant components. Input selection for classification and regression problems is a task-specific form of dimensionality reduction. Visualization of high-dimensional data requires mapping to lower dimension[75][64].

5.1 INTRODUCTION

A global data model is built by a combination of many simple local models (generally linear). It can be justified because of the following facts[65][75][80] -

(1) A global data model may actually consists of separate manifolds which may or may not be connected together in one piece, i.e. it may be clustered.

(2) The intrinsic dimensionality of the data may vary along the manifold.

(3) Any differentiable function becomes approximately linear in a sufficiently small region around a point.

It may happen that the global data might be highly nonlinear and require the whole data space to embed it, but the individual parts of the whole data may require simple and linear models. It may cause the following drawbacks[67][65] -

(1) A large data set is required to fit a large number of parameters.

(2) Training becomes difficult because the error function is likely to have a lot of local
minima due to the high flexibility of the model.

The global data is split into simple parts which can be trained easily. Local dimensionality reduction is a faster approach with little data and no (or few) local minima. Hence there is no single, global dimensionality reduction mapping (from data space to a single low-dimensional space) because each mapping has its low-dimensional space with its own dimension. And thus there is no single, global reconstruction mapping as described in [64][66][75].

5.1.1 Requirements for Local Dimensionality Reduction Models

(1) Simple dimensionality reduction models as building blocks, usually distributed around the space and each one having a limited reach (hence the locality)[66][75].

(2) Method to determine the dimensionality of each component.

(3) Clustering strategy in which a weight-rule is to be defined which assigns a weight to each component.

(5) Approach to train both the local models and the clustering.

The weight-assignment can be[65][75][80]

(1) **Hard Weight Assignment** : A single component receives all the weight-responsibility and the rest receive no responsibility at all.

(2) **Soft Weight Assignment** : The weight-responsibility is distributed among all components as a partition of unity. When training is done, a given data point will result in an update of all components and when dimensionality is reduced, the reduced-dimension representative will be the average of the local reduced-dimension representatives weighted by the respective responsibilities.

It has been observed that the soft weight-assignation of responsibilities is better than the hard assignation because it is a probabilistic framework based upon the results of the optimality of the mean of a distribution to represent the whole distribution.

From a probalistic point of view, the concept of local models and weight assignation can be expressed as a mixture (of latent variable models). The training criterion can be log-likelihood rather than reconstruction error, since the probability model attempts to model the noise as well as the underlying manifold[75][80].
The Lloyd algorithm iteratively computes the partition based on two criteria that insure minimum average distortion. These two criteria are[17][65] -

1. VQ centers lie at the generalized centroids of the quantizer regions.

2. the VQ region boundaries are surfaces that are equi-distant (in terms of the distortion function) from adjacent VQ centers.

The simplest way to construct the partition is to build a VQ based on Euclidean distance. This can be done by the generalized Lloyd algorithm. The trained quantizer consists of a set of $Q$ reference vectors $r^{(i)} = r(i), i = 1, \ldots, Q$ and corresponding regions $R^{(i)}$. The placement of the reference vectors and the definition of the regions satisfy Lloyd’s optimality conditions[65] -

1. Each region, $R^{(i)}$ corresponds to all $x \in \mathbb{R}^n$ that lie closer to $r^{(i)}$ than to any other reference vector. Mathematically [65],

   $$ R^{(i)} = \{ x \mid d_E(x,r^{(i)}) < d_E(x,r^{(j)}), \text{ for all } j \neq i \} $$

   where $d_E(a,b)$ is the Euclidean distance between $a$ and $b$. Thus, a given $x$ assigned to its nearest neighbor $r(i)$.

2. Each reference vector $r^{(i)}$ is placed at the centroid of the corresponding region $R^{(i)}$. For Euclidean distance, the centroid is the mean $r^{(i)} = E[ x \mid x \in R^{(i)} ]$.

For Euclidean distance, the regions are connected and the convex sets are known as Voronoi cells. The $m$-dimensional encoding of the original vector $x$ is thus given in two parts[65] -

(a) the index of the Voronoi region that the vector lies in.

(b) the local coordinates of the point with respect to the centroid, in the basis of the $m$ leading eigenvectors of the corresponding covariance.

If $x \in R^{(i)}$, the local coordinates are [65]

$$ z = (e^{(i)}_1 \cdot (x - r^{(i)}), \ldots, e^{(i)}_m \cdot (x - r^{(i)})) $$

The decoded vector is given by [65]
The mean squared reconstruction error incurred is

\[ \xi_{\text{recon}} = E[\| x - \hat{x} \|^2] \]

The search and storage requirements for computing the partition can be streamlined by constraining the structure of the VQ. These constraints can compromise performance relative to what could be achieved with a standard, or unconstrained, VQ with the same number of regions[65][75].

5.2 VECTOR QUANTIZATION

A vector quantizer is a system for mapping a sequence of continuous or discrete vectors into a digital sequence suitable for communication or for storage in a digital channel. The goal of a vector quantizer based system is to reduce the bit rate so as to minimize communication channel capacity or digital storage memory requirements while maintaining the necessary fidelity of the data[9][10]. Image coding and speech coding are the most important applications of data compression.

Both of these applications need considerable compression with sufficiently sophisticated coders. During the past several years many design algorithms have been developed for a variety of vector quantizers and the performance of these codes has been studied for speech waveforms, speech linear predictive parameter vectors, images, and several simulated random processes[24].

Vector Quantization is suitable for pattern recognition due to ease of implementation and high accuracy[28]. The VQ is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and it can be represented by the centroid called codeword. The collection of all codewords consists of the corresponding codebook for the known speaker. Figure 5.1 shows a conceptual diagram to illustrate this recognition process. In the figure only two clusters and two dimensions of the acoustic space are shown. The circles refer to the acoustic vectors from cluster-1 while the triangles are from the cluster-2 [10][13][17].

In the training phase, a cluster specific VQ codebook is generated for each known cluster by clustering its training acoustic vectors. The result codewords are shown by black circles and black triangles for clusters 1 and 2, respectively[13].
The distance from a vector to the closest codeword is called distortion. In the recognition phase, an input utterance of an unknown voice is vector-quantized using each trained codebook and the total distortion is computed. The speaker corresponding to the VQ codebook with smallest total distortion is identified[17].

In 1980, Linde, Buzo and Gray (LBG) proposed a VQ design algorithm based on a training sequence. The use of a training sequence bypasses the need for multi-dimensional integration. A VQ designed using this algorithm is known as LBG-VQ[17][13].

5.2.1 VQ Design Problem

Given a vector space with its statistical properties known, given a distortion measure, and given the number of code-vectors, find a codebook and a partition, which result in the smallest average distortion[13][17][65].

Let there be a training sequence consisting of M source vectors:

\[ T = \{ x_1, x_2, x_3, \ldots, x_M \} \]

It is assumed that all the statistical properties of the source are captured by the training sequence. It is assumed that the source vectors are k-dimensional.

\[ x_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,k}), \quad m = 1, 2, 3, \ldots, M \]

Let N be the number of code-vectors and let

\[ C = \{ c_1, c_2, c_3, \ldots, c_N \} \]

represents the codebook. Each code-vector is k-dimensional, e.g.

\[ c_n = (c_{n,1}, c_{n,2}, \ldots, c_{n,k}), \quad n = 1, 2, 3, \ldots, N \]

Let \( S_n \) be the encoding region associated with code-vector \( c_n \) and let

\[ P = \{ S_1, S_2, \ldots, S_N \} \]

denote the partition of the space. If the source vector \( x_m \) is in the encoding region \( S_n \), then its approximation (denoted by \( G(x_m) \)) is \( c_n \).

\[ G(x_m) = c_n, \quad \text{if} \quad x_m \in S_n \]
The squared-error distortion measure is given by

\[ D_{ave} = \frac{1}{Mk} \sum_{m=1}^{M} \| x_m - G(x_m) \|^2 \]

where \[ \| e \|^2 = e_1^2 + e_2^2 + \ldots + e_k^2 \]

Hence for given \( T \) and \( N \), find \( C \) and \( P \) such that \( D_{ave} \) is minimized.

**Optimality Criteria**: Since \( C \) and \( P \) are a solution to the above minimization problem, it satisfies the following two criteria[17] -

- Nearest Neighbourhood Condition[17][65]

  \[ S_n = \{ x : \| x - c_n \|^2 \leq \| x - c_{n'} \|^2 \ \forall n' = 1, 2, 3, \ldots N \} \]

  This condition states that the encoding region \( S_n \) should consists of all vectors that are closer to \( c_n \) than any of the other code-vectors.

- Centroid Condition[17][65]

---

**Figure 5.1 VQ Codebook Formation (after[13])**
This condition describes that the code-vector $c_n$, should be the average of all those training vectors that are in encoding region $S_n$.

$$
c_n = \frac{\sum_{x_m \in S_n} x_m}{\sum_{x_m \in S_n} 1}, \quad n = 1, 2, 3, \ldots, N
$$

The LBG-VQ algorithm is an iterative algorithm, which alternatively solves the above two optimality criteria. The algorithm requires an initial codebook $C^{(0)}$. This initial codebook is obtained by the splitting method. The initial code-vector is set as the average of the entire training sequence. This code vector is then split into two. The iterative algorithm is run with these two vectors as the initial codebook. The final two code vectors are split into four and the process is repeated until the desired number of code-vectors is obtained[17][65].

Figure 5.2 shows the flow-chart for the LBG-VQ algorithm. The “cluster vectors” is the nearest-neighbour search procedure which assigns each training vector to a cluster associated with the closest codeword. The “find centroids” is the centroid update procedure. Compute $D_{\text{ave}}$ sums the distances of all training vectors in the nearest-neighbour search so as to determine whether the procedure has converged.

**Benefits of VQ Representation**

(i) **Efficient Storage of Relevant Information** : The reduction of dimensionality of speech patterns and training sets length, is very important for optimizing the storage and transmission of signals.

(ii) **Reduced Training Sets Generation** : The generation of reduced codebooks (training sets) for representing speech information has a direct implication in fast response retrieval and memory constraints of speech recognizers.

(iii) **Simplified Labeling Process** : The process of labeling can be easily done using VQ techniques. The association of a phonetic class directly to each codeword can be equivalent to assigning a phonetic label to each speech frame.

**Disadvantages of VQ Representation**

(i) **Distortion** : VQ coding of information, carry an implicit distortion of the original signal due to quantization constraints. As the number of bits used to represent a pattern decrease, the quantization error increases.
(ii) **Codebook Storage**: the selection of an adequate number of codewords is often a difficult trade-off to establish between accuracy and speed-storage constraints.

### 5.2.2 VQPCA and VQ_EM_FA

Despite the wide use of PCA, its transformation is crippled by its reliance on second order statistics. Though uncorrelated, the principal components can be highly statistically dependent. When this is the case, PCA fails to find the most compact description of the data. Geometrically, PCA models the data as a hyperplane embedded in the ambient space. If the data components have non-linear dependancies, PCA requires a larger-dimensional representation than that by a non-linear technique.
Because of this, the non-linear alternatives to PCA have been developed[65][75].

The vector quantization PCA algorithm uses a vector quantization algorithm to produce a hard partition of the data space (a Voronoi tessellation). In each of the Voronoi cells, a separate PCA is fitted, whose mean and covariance matrix are calculated from the data points in that cell. Assigning data points to centroids by any distortion measure such as Euclidean distance or weighted Euclidean measure and projection on its local space are being observed. This model can be applied to the dimensionality reduction of the speech data[17][65][75].

The VQPCA algorithm directly minimizes the reconstruction error and it is optimal for dimensionality reduction. The computation of the partition is a generalised Lloyd algorithm for a vector quantizer that uses the reconstruction error as the VQ distortion function[17][65][75].

In the VQ_EM_FA algorithm, the covariance matrix for each region is approximated on the basis of number of factors to be used. It is assumed that the common factors and the specific variances are normally distributed. Maximum likelihood estimates of the factor loadings and specific variances are obtained by the expectation-maximization algorithm for each region.

\[
\text{Pseudocode of the VQPCA algorithm of Kambhatla and Leen (1997) (after[65])}
\]

\[
\begin{align*}
\text{initialise} & \quad \{ \mu_m \}_{m=1}^M \leftarrow \text{random subset of the data points} \\
& \quad \Sigma_m \leftarrow I \quad \text{for each } m = 1, \ldots, M \\
\text{repeat} & \quad \text{Assign each data point to the cluster that best reconstructs it} \\
& \quad \text{for } m = 1, \ldots, M \\
& \quad \mu_m \leftarrow \text{mean of data points assigned to cluster } m \\
& \quad \Sigma_m \leftarrow \text{covariance matrix of data points assigned to cluster } m \\
& \quad U_m \leftarrow \text{first principal components of } \Sigma_m \\
\text{end} & \quad \text{Update centroids} \\
\text{until} & \quad \text{the fractional change in average reconstruction error is small} \\
\text{return} & \quad \{ \mu_m, U_m \}_{m=1}^M
\end{align*}
\]
5.3 SUMMARY

In this chapter, we have described VQ, a piece-wise constant modelling technique which partitions the input space into a set of regions and approximates each region with a reference vector. Then we have described local linear algorithms for dimension reduction- VQPCA and VQ_EM_FA. Both of these algorithms partition the input space into disjoint regions using a VQ. They then build local low dimensional coordinate systems using a PCA and FA (expectation maximization based). Thus they form a local linear model of the data.

In the next chapter we will implement VQPCA and VQ_EM_FA on various machine learning datasets and compare the recognition results with global dimension reduction methods.

initialise

\{ \mu_m \}_{m=1}^{M} \leftarrow \text{random subset of the data points} \hspace{2cm} \text{Centroids}

\Sigma_m \leftarrow \text{I} \hspace{2cm} \text{Covariance}

repeat

Assign each data point to the cluster that best reconstructs it \hspace{2cm} \text{Hard assignment}

\text{for } m = 1, \ldots, M \hspace{2cm} \text{For each cluster}

\mu_m \leftarrow \text{mean of data points assigned to cluster } m \hspace{2cm} \text{Update centroids}

\Sigma_m \leftarrow \text{covariance matrix of data points assigned to cluster } m \hspace{2cm} \text{Update covariance}

F_m \leftarrow \text{first Factor Loadings of } \Sigma_m \hspace{2cm} \text{Update local FA directions}

end

until the fractional change in average reconstruction error is small

return \{ \mu_m, F_m \}_{m=1}^M

Pseudocode of the proposed VQ_EM_FA algorithm
In this section we have described various recognition experiments where different dimensionality reduction methods are evaluated on different datasets. We reduce the dimensionality of the feature space from $D$ to $d$ (where $d < D$). Recognition experiments with the original feature datasets are described to provide baseline results.

In order to evaluate the dimensionality reduction methods, different datasets were used. Brief details of these datasets is described in the following section 6.1.1. Each dataset has been divided into two sets of data. One set of data is used for training and another for testing. In all of the following experiments, the training dataset is used for computing transformation required in these dimensionality reduction methods. The test dataset is used only to compute the recognition performance of the recognizer with different reduced feature sets.

### 6.1.1 Details of the datasets used

To evaluate the dimensionality reduction methods, recognition accuracy was computed for 5 different datasets. A higher dimensionality dataset, obtained by decomposing a standard image, was also used to compare different techniques. The reasons for choosing these specific datasets are:

- they are available for public download from various ftp sites.
- all the above datasets have been studied earlier by other researchers making the results easy to compare and reproduce [86][87][88].
Brief description of the machine learning databases are used in our experiments for dimensionality reduction experiments is given below.

(1) Peterson Barney (Vowel formant frequency database): This database consists of frequency and amplitude of 1520 utterances, which is resulted from the 2 repetitions of 10 vowels.

Number of Attributes (Dimension) : 4

Attribute Information: F0, F1, F2 and F3 represent frequency and amplitude of different utterances.

Number of Instances (Vectors) : 1520

It consists of 2 repetitions of 76 instances of 10 vowels. The 76 instances per vowel contain data for 33 male, 28 female and 15 children.

Number of Classes : 10

These classes represent 10 vowels as IY, IH, EH, AE, AH, AA, AO, UH, UW and ER.

(origin: ftp.cis.upenn.edu/pub/ldc/pb.data.tar.z Creator: Peterson Berney)

(2) GLASS Identification Database: This database consists of data used to classify types of glasses. It was used in determining whether the glass was a type of ‘float’ glass or not.

Number of Attributes (Dimension) : 10 (including an Id#)

Attribute Information : Id number, RI, Na, Mg, Al, Si, K, Ca, Ba and Fe. The attributes are the weight present in corresponding oxide.

Number of Instances (Vectors) : 163

Number of Classes : 2

(1) Window glass-float processed : 87

(2) Window glass-non float processed : 76

(origin: ftp.ics.uci.edu/pub/machine-learning-databases/glass Creator: B.German)
(3) **WINE Recognition Database**: This database consists of data which are the results obtained from a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.

Number of Attributes (Dimension): 13

These attributes are: Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Non-flavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280 /OD315 of diluted wines and Proline.

Number of Classes: 3

Number of Instances (Vectors):

- Class 1: 59
- Class 2: 71
- Class 3: 48


(4) **IRIS Plants Database**: This database contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the later are not linearly separable from each other. This dataset has been be used in many pattern recognition problems.

Number of Attributes (Dimension): 4

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

Number of Classes: 3

1. Iris Setosa
2. Iris Versicolour
3. Iris Virginica
Number of Instances (Vectors): 150 (50 in each of the three classes)


(5) **Deterding Database (Vowel Recognition)**: It is a dataset of speaker independent recognition of the eleven steady state vowels of British English using a specified training set of lpc derived log area ratios. The vowels are indexed by integers 0-10. For each utterance, there are ten floating-point input values, with array indices 0-9. The vowels are the following: hid, hId, hEd, hAd, hYd, had, hOd, hod, hUd, hud, hed.

Number of Attributes (Dimension): 10

Number of Classes: 11

Number of Instances (Vectors): 48 (train), 42 (test)


(6) **WBC (Wisconsin Breast Cancer) Database**: This dataset contains instances obtained by Dr Wolberg of the University of Wisconsin Hospitals, Madison, USA by the multisurface method of pattern separation for medical diagnosis applied to breast cytology. Each instance has one of the 2 possible classes: benign or malignant.

Number of Attributes (Dimension): 30

Number of Classes: 2

Number of Instances (Vectors):

Class 1: 356

Class 2: 212

(origin: ftp.ics.uci.edu/pub/machine-learning-databases Creator: Dr W.H. Wolberg)
6.1.2 Dimensionality Reduction using Principal Component Analysis

Principal Component Analysis (PCA) is the most widely used dimensionality reduction technique in practice. This is because of its conceptual simplicity, its analytical properties and also the fact that relatively efficient algorithms (of polynomial complexity) exist for its computation. In signal processing it is known as Karhunen-Loeve transform [38][39][75].

**Approach**

PCA is used to compute a linear transformation which projects the original $D$-dimensional feature space on a $d$-dimensional subspace (where $d < D$).

1. The total covariance matrix $\mathbf{T}$ is computed from the training data set using the following equation:

$$
\mathbf{T} = \frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} (\mathbf{x}_{kn} - \mu) (\mathbf{x}_{kn} - \mu)^{\intercal}
$$

where

- $\mathbf{x}_{kn}$: these are the D-dimensional vectors (or instances) in the training set, $k = 1, \ldots, K$ and $n = 1, \ldots, N$. Number of training vectors in each of the $K$ classes is assumed to be the same ($= N$) for simplifying the presentation.
- $\mu$: is the overall mean given by

$$
\mu = \frac{1}{KN} \sum_{k=1}^{K} \sum_{n=1}^{N} \mathbf{x}_{kn}
$$

2. The linear transformation is given by the matrix $\mathbf{U}$, where $\mathbf{U}$ is a $D \times d$ unitary matrix whose columns are the eigenvectors corresponding to the $d$ largest eigenvalues of the total covariance matrix $\mathbf{T}$. When no dimensionality reduction is performed, (when $d = D$), this transformation amounts to a rotation in the feature space.

3. Recognition results are computed using the test dataset as a function of dimensionality $d$. We have used Mahalanobis distance measure to compute the distance
of a feature vector from a given class. Here, a feature vector \( y \) is classified to \( j \)-th class if the distance \( d_j(y) \) is less than the other distances \( d_i(y), i = 1, \ldots, K \). The distance \( d_j(y) \) is computed as

\[
d_j(y) = (y - \mu_j)^T \Sigma_j^{-1} (y - \mu_j)
\]

where

\( \mu_i \) is the mean vector of the class \( i \) and \( \Sigma_i \) is the covariance matrix. We have used the full covariance matrix in our experiments.

Some intermediate results i.e. transformation matrix, eigenvalues, eigenvectors and recognition performance matrix after testing are given in section of the Appendix-B.

### Results

![PCA Recognition Accuracy vs Dimension (GLASS Dataset)](#)

![PCA Eigenvalues vs Dimension Plot (GLASS Dataset)](#)

![PCA Recognition Accuracy vs Dimension (WINE Dataset)](#)

![PCA Eigenvalues vs Dimension Plot (WINE Dataset)](#)
3. IRIS dataset

4. PB dataset

5. Deterding dataset

EXPERIMENTS

PCA Recognition Accuracy v/s Dimension (IRIS Dataset)

Dimension

Recognition Accuracy %

0

20

40

60

80

100

120

Dimension

Eigen Values

0

1e+0

2e+0

3e+0

4e+0

5e+0

6e+0

PCA Eigenvalues v/s Dimension Plot (IRIS Dataset)

Dimension

PCA Recognition Accuracy v/s Dimension (PB Dataset)

Dimension

Recognition Accuracy %

0

20

40

60

80

100

Dimension

Eigen Values

0

1e+5

2e+5

3e+5

4e+5

5e+5

6e+5

PCA Eigenvalues v/s Dimension Plot (PB Dataset)

Dimension

PCA Recognition Accuracy v/s Dimension (Deterding Dataset)

Dimension

Recognition Accuracy %

0

20

40

60

80

100

Dimension

Eigen Values

0

5e-3

1e-2

2e-2

2e-2

PCA Eigenvalues v/s Dimension Plot (Deterding Dataset)

Dimension
6. BCW dataset

6.1.3 Dimensionality Reduction using Linear Discriminant Analysis

LDA finds the optimum transformation matrix as to preserve most of the information that can be used to discriminate between the different classes.[2][38][39]

Approach

LDA is used to compute a linear transformation which projects the original $D$-dimensional feature space on a $d$-dimensional subspace (where $d < D$).

1. In LDA, within-class covariance matrix ($W$) and between-class covariance matrix ($B$) are used to formulate the criterion of class separability. These matrices are symmetric and computed from the training set by using the following

$$B = \frac{1}{K} \sum_{k=1}^{K} (\mu_k - \mu)(\mu_k - \mu)^T$$

$$W = \frac{1}{K} \sum_{k=1}^{K} W_k$$

where $\mu_k$ and $W_k$ are the mean vector and covariance matrix of the $k^{th}$ class, respectively, and $\mu$ is the overall mean. These are given by
\[ \mu_k = \frac{1}{N} \sum_{n=1}^{N} x_{kn} \]

\[ \mathbf{W}_k = \frac{1}{N} \sum_{n=1}^{N} ( x_{kn} - \mu ) ( x_{kn} - \mu_k )^T \]

\[ \mu = \frac{1}{K} \sum_{k=1}^{K} \mu_k \]

where \( x_{kn} \) is the \( n \)-th training pattern from the \( k \)-th class. The number of training patterns in each of the \( K \) classes is assumed to be the same (\( =N \)) for simplifying the presentation.

2. The linear transformation is given by the matrix \( \mathbf{U} \), where \( \mathbf{U} \) is a \( D \times d \) matrix whose columns are the eigenvectors corresponding to the \( d \) largest eigenvalues of the matrix \( \mathbf{W}^{-1} \mathbf{B} \).

3. Reduced feature sets of different sizes are computed using this method and the recognition performance of the recognizer is evaluated on the test dataset for each of these reduced feature sets. Recognition results are computed using the test dataset as a function of dimensionality \( d \). We have used Mahalanobis distance measure to compute the distance of a feature vector from a given class. Here, a feature vector \( y \) is classified to \( j \)-th class if the distance \( d_j(y) \) is less than the other distances \( d_i(y), i = 1, \ldots, K \). The distance \( d_i(y) \) is computed as

\[ d_i(y) = (y - \mu_i)^T \Sigma_i^{-1} (y - \mu_i) \]

where

\( \mu_i \) is the mean vector of the class \( i \) and \( \Sigma_i \) is the covariance matrix. We have used the full covariance matrix in our experiments.

Some intermediate results i.e. transformation matrix, eigenvalues, eigenvectors and recognition performance matrix after testing are given in section of the Appendix-B.
Results

1. GLASS dataset

   ![Graph showing LDA Recognition Accuracy vs Dimension (GLASS Dataset)]

   ![Graph showing LDA Eigen Values vs Dimension Plot (GLASS Dataset)]

2. WINE dataset

   ![Graph showing LDA Recognition Accuracy vs Dimension (WINE Dataset)]

   ![Graph showing LDA Eigen Values vs Dimension Plot (WINE Dataset)]

3. IRIS dataset

   ![Graph showing LDA Recognition Accuracy vs Dimension (IRIS Dataset)]

   ![Graph showing LDA Eigen Values vs Dimension Plot (IRIS Dataset)]
4. PB dataset

LDA Recognition Accuracy v/s Dimension (PB Dataset)

LDA Eigenvalues v/s Dimension Plot (PB Dataset)

5. Deterding dataset

LDA Recognition Accuracy v/s Dimension (Deterding Dataset)

LDA Eigenvalues v/s Dimension Plot (Deterding Dataset)

6. BCW dataset

LDA Recognition Accuracy v/s Dimension (BCW Dataset)

LDA Eigenvalues v/s Dimension Plot (BCW Dataset)
### 6.1.3 Dimensionality Reduction using Expectation-Maximization Factor Analysis

In maximum likelihood factor analysis, the generative model as described in [67][79][83] is given by the following equation –

\[
X = \hat{\Phi} Z + u
\]

........................ (1)

where

- \(X\) is a \(p\)-dimensional real-valued data vector
- \(Z\) is \(k\)-dimensional vector of real-valued factors. The factors \(z\) are assumed to be normally distributed \(N(0,I)\), zero-mean independent normal with unit variance.
- \(\hat{\Phi}\) is a factor-loading matrix.
- \(u\) is a \(p\)-dimensional random variable distributed \(N(0,\Theta)\), where \(\Theta\) is a diagonal matrix.

In the factor analysis model, the value of \(k\) is generally much smaller than \(p\).

The \(k\) factors are the informative projections of the data. These are similar to the principal components in PCA. When the \(\hat{\Phi}\) and \(\Theta\) is given, the expected value of the factors can be computed through the linear projections –

\[
E(Z|X) = \hat{\alpha} X
\]

........................ (2)
where $\hat{a} = \hat{E}'(\theta + \hat{E} \hat{E}')^{-1}$

The above value of $\hat{a}$ is resulted from the joint normality of data and factors.

$$P \left( \begin{bmatrix} X \\ Z \end{bmatrix} \right) = N \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Lambda \Lambda' + \Psi & \Lambda' \\ \Lambda & I \end{bmatrix} \right)$$

.......................... (3)

Here $\theta$ is diagonal, the $p \times p$ matrix $(\theta + \hat{E} \hat{E}')$ can be efficiently inverted using the matrix inversion lemma –

$$(\theta + \hat{E} \hat{E}')^{-1} = \theta^{-1} - \theta^{-1} \hat{E} (I + \hat{E}' \theta^{-1} \hat{E})^{-1} \hat{E}' \theta^{-1}$$

where $I$ is the $k \times k$ identity matrix. Proof of this lemma is given in Appendix -1.

In EM method, it is necessary to compute the second moment of the factors as –

$$E(ZZ'|X) = \text{Var}(Z|X) + E(Z|X) E(Z|X)'$$

$$= I - \hat{a} \hat{E} + \hat{a} X X' \hat{a}'$$

..............................(4)

The above equation provides a measure of uncertainty in the factors. This is a quantity, which has no analogue in PCA.

The expectations (2) and (4) above are the basis of the EM algorithm for maximum likelihood factor analysis.[79][83]

**E-step:** Compute $E(z|x_i)$ and $E(zz'|x_i)$ for each data point $x_i$ for the given values of $\hat{E}$ and $\theta$.

**M-step:**

$$\Lambda^{\text{new}} = \left( \sum_{i=1}^{n} X_i E(Z|X_i)' \right) \left( \sum_{i=1}^{n} E(ZZ'|X_i) \right)^{-1}$$

$$\Psi^{\text{new}} = \frac{1}{n} \text{diag} \left\{ \sum_{i=1}^{n} X_i X_i' - \Lambda^{\text{new}} E(Z|X_i) X_i' \right\}$$

where the diag operator sets all the off-diagonal elements of a matrix to zero.
Approach

1. The factor loading matrix $\hat{E}_{new}$ is computed by using the above equations on the training dataset.

2. The linear transformation is given by the matrix $U'$, where $U$ is a $D \times d$ unitary matrix whose columns are the eigenvectors corresponding to the $d$ largest eigenvalues of the factor loading matrix $\hat{E}_{new}$.

3. Recognition results are computed using the test dataset as a function of dimensionality $d$. We have used Mahalanobis distance measure to compute the distance of a feature vector from a given class. Here, a feature vector $y$ is classified to $j$-th class if the distance $d_j(y)$ is less than the other distances $d_i(y)$, $i = 1, ..., K$. The distance $d(y)$ is computed as

$$d_i(y) = (y - \mu_i)^T \Sigma_i^{-1} (y - \mu_i)$$

where

$\mu_i$ is the mean vector of the class $i$ and $\Sigma_i$ is the covariance matrix. We have used the full covariance matrix in our experiments.

Some intermediate results i.e. transformation matrix, eigenvalues, eigenvectors and recognition performance matrix after testing are given in section of the Appendix-B.

Results

1. GLASS dataset
2. WINE dataset

3. IRIS dataset

4. PB dataset
5. Deterding dataset

6. BCW dataset

Comparison:

Table 6.1 Baseline results with the original datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Dimensions</th>
<th>Recognition Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PCA</td>
</tr>
<tr>
<td>1 PB</td>
<td>4</td>
<td>51.44</td>
</tr>
<tr>
<td>2 GLASS</td>
<td>9</td>
<td>85.3</td>
</tr>
<tr>
<td>3 WINE</td>
<td>13</td>
<td>75</td>
</tr>
<tr>
<td>4 IRIS</td>
<td>4</td>
<td>98</td>
</tr>
<tr>
<td>5 Deterding</td>
<td>10</td>
<td>94.50</td>
</tr>
<tr>
<td>6 BCW</td>
<td>30</td>
<td>79.75</td>
</tr>
</tbody>
</table>
Results of IRIS, WINE, GLASS and BCW datasets for PCA and LDA methods are comparable with results reported by H. Brunzell, J. Eriksson [87]. Results of PB dataset for PCA and LDA are comparable with the results reported by K.K. Paliwal, M. Bacchiani, Y. Sagisaka [88]. Results for EM_FA method are obtained by our investigation on these datasets.

**Table 6.2 IRIS dataset**

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Recognition Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCA</td>
</tr>
<tr>
<td>4</td>
<td>98.00</td>
</tr>
<tr>
<td>3</td>
<td>98.67</td>
</tr>
<tr>
<td>2</td>
<td>96.67</td>
</tr>
<tr>
<td>1</td>
<td>93.33</td>
</tr>
</tbody>
</table>

**Table 6.3 PB dataset**

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Recognition Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCA</td>
</tr>
<tr>
<td>4</td>
<td>51.84</td>
</tr>
<tr>
<td>3</td>
<td>51.05</td>
</tr>
<tr>
<td>2</td>
<td>38.42</td>
</tr>
<tr>
<td>1</td>
<td>23.28</td>
</tr>
</tbody>
</table>

**Table 6.4 GLASS dataset**

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Recognition Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCA</td>
</tr>
<tr>
<td>9</td>
<td>82.28</td>
</tr>
<tr>
<td>8</td>
<td>87.12</td>
</tr>
<tr>
<td>7</td>
<td>84.66</td>
</tr>
<tr>
<td>6</td>
<td>82.82</td>
</tr>
<tr>
<td>5</td>
<td>83.44</td>
</tr>
<tr>
<td>4</td>
<td>82.82</td>
</tr>
<tr>
<td>3</td>
<td>76.07</td>
</tr>
<tr>
<td>2</td>
<td>73.62</td>
</tr>
<tr>
<td>1</td>
<td>49.69</td>
</tr>
</tbody>
</table>
### Table 6.5 WINE dataset

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Recognition Accuracy %</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCA</td>
<td>LDA</td>
</tr>
<tr>
<td>13</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>12</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>11</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>10</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>9</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>8</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>7</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>6</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>5</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>4</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>3</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>2</td>
<td>75</td>
<td>97.73</td>
</tr>
<tr>
<td>1</td>
<td>75</td>
<td>69.32</td>
</tr>
</tbody>
</table>

### Table 6.6 Deterding dataset

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Recognition Accuracy %</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCA</td>
<td>LDA</td>
</tr>
<tr>
<td>10</td>
<td>94.50</td>
<td>94.50</td>
</tr>
<tr>
<td>9</td>
<td>91.85</td>
<td>91.47</td>
</tr>
<tr>
<td>8</td>
<td>89.96</td>
<td>86.55</td>
</tr>
<tr>
<td>7</td>
<td>86.74</td>
<td>77.27</td>
</tr>
<tr>
<td>6</td>
<td>83.71</td>
<td>72.15</td>
</tr>
<tr>
<td>5</td>
<td>79.54</td>
<td>53.78</td>
</tr>
<tr>
<td>4</td>
<td>68.18</td>
<td>43.18</td>
</tr>
<tr>
<td>3</td>
<td>51.13</td>
<td>32.57</td>
</tr>
<tr>
<td>2</td>
<td>43.18</td>
<td>19.88</td>
</tr>
<tr>
<td>1</td>
<td>29.54</td>
<td>14.58</td>
</tr>
</tbody>
</table>
6.2 RECOGNITION ACCURACY ON RECONSTRUCTED DATASETS ON REDUCED DIMENSIONALITY

As we have described in section 6.1.2 that the linear transformation is given by the matrix $U'$, where $U$ is a $D \times d$ unitary matrix whose columns are the eigenvectors corresponding to the $d$ largest eigenvalues of the total covariance matrix $\mathbf{T}$. The linear transformation projects the original $D$-dimensional feature space on a $d$-dimensional subspace (where $d < D$). Given that $\mu$ is the overall mean, the reconstructed vector from $x$ can be computed as

$$\hat{x} = U U^T (x - \mu) + \mu$$

Graphs showing the recognition accuracy and mean square error on the reconstructed datasets with reduced dimensionality are shown below for different datasets.
A. Principal Component Analysis (PCA)

1. GLASS dataset

![Recognition Accuracy vs Dimension (GLASS Dataset)](image1)

![MSE vs Dimension Plot (GLASS Dataset)](image2)

2. WINE dataset

![Recognition Accuracy vs Dimension (WINE Dataset)](image3)

![MSE vs Dimension Plot (WINE Dataset)](image4)

3. IRIS dataset

![Recognition Accuracy vs Dimension (IRIS Dataset)](image5)

![MSE vs Dimension Plot (IRIS Dataset)](image6)
4. PB dataset

![Graph showing Recognition Accuracy vs Dimension (PB Dataset)]

![Graph showing MSE vs Dimension (PB Dataset)]

5. Deterding dataset

![Graph showing Recognition Accuracy vs Dimension (Deterding Dataset)]

![Graph showing MSE vs Dimension (Deterding Dataset)]

6. BCW dataset

![Graph showing Recognition Accuracy vs Dimension (BCW Dataset)]

![Graph showing MSE vs Dimension (BCW Dataset)]
B. Linear Discriminant Analysis (LDA)

1. GLASS dataset

![GLASS dataset graphs]

2. WINE dataset

![WINE dataset graphs]

3. IRIS dataset

![IRIS dataset graphs]
4. PB dataset

![LDA Recognition Accuracy vs Dimension (PB Dataset)](image1)

![LDA MSE vs Dimension Plot (PB Dataset)](image2)

5. Deterding dataset

![LDA Recognition Accuracy vs Dimension (Deterding Dataset)](image3)

![LDA MSE vs Dimension Plot (Deterding Dataset)](image4)

6. BCW dataset

![LDA Recognition Accuracy vs Dimension (BCW Dataset)](image5)

![LDA MSE vs Dimension Plot (BCW Dataset)](image6)
C. Factor Analysis (EM_FA) expectation-maximization based

1. GLASS dataset

2. WINE dataset

3. IRIS dataset
4. PB dataset

![EM_FA Recognition Accuracy v/s Dimension (PB Dataset)](image)

![EM_FA MSE v/s Dimension Plot (PB Dataset)](image)

5. Deterding dataset

![EM_FA Recognition Accuracy v/s Dimension (Deterding Dataset)](image)

![EM_FA MSE v/s Dimension Plot (Deterding Dataset)](image)

6. BCW dataset

![EM_FA Recognition Accuracy v/s Dimension (BCW Dataset)](image)

![EM_FA MSE v/s Dimension Plot (BCW Dataset)](image)
6.3 VQPCA and VQ_EM_FA FOR LOCAL DIMENSIONALITY REDUCTION

PCA and EM_FA techniques of dimensionality reduction have been implemented using vector quantization for local dimensionality reduction on various machine-learning databases.

For local dimensionality reduction, the VQ_EM_FA and VQ_PCA algorithms use a vector quantization algorithm to produce a hard partition of the data space (a Voronoi tessellation). In each of the Voronoi cells, a separate EM_FA or PCA is fitted, whose mean and covariance matrix are calculated from the data points in that cell. Assigning data points to centroids by any distortion measure such as Mahalanobis distance and projection on its local space are being observed. Global data, split into 2, 4 and 8 Voronoi cells (partitions) using VQ and then EM_FA or PCA has been implemented for local dimensionality reduction on each Voronoi cell (partition) and the classifier is trained and tested on different databases. VQPCA and VQ_EM_FA algorithms are given on pages numbers 60-61 of the Chapter-5. Recognition accuracy was calculated after reconstructing the data sets after reducing the dimensionality locally. The dimensionality on each partition has been reduced to 1 from the highest dimension of each dataset for comparison purpose.

Results

1. IRIS dataset

![Recognition Accuracy vs Num of Levels (IRIS Dataset)](image)
2. PB dataset
3. Deterding dataset
6.5 COMPARING PCA v/s VQPCA and FA v/s VQ_EM_FA

The following tables compare the recognition results of PCA with VQPCA and EM_FA with VQ_EM_FA methods when the dimensionality of various datasets is reduced to 1. It shows that when number of levels (partitions) in VQ based methods are increased, the recognition accuracy increases.

Table 6.8 PCA and VQPCA when global data is split into different levels (partitions)

<table>
<thead>
<tr>
<th>Datasets</th>
<th>PCA</th>
<th>VQ_PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>IRIS</td>
<td>74.66</td>
<td>96</td>
</tr>
<tr>
<td>PB</td>
<td>23.28</td>
<td>28.95</td>
</tr>
<tr>
<td>Deterding</td>
<td>25.56</td>
<td>56.06</td>
</tr>
</tbody>
</table>

Table 6.9 EM_FA and VQ_EM_FA when global data is split into different levels (partitions)

<table>
<thead>
<tr>
<th>Datasets</th>
<th>EM_FA</th>
<th>VQ_EM_FA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>IRIS</td>
<td>74.66</td>
<td>94.67</td>
</tr>
<tr>
<td>PB</td>
<td>23.68</td>
<td>26.45</td>
</tr>
<tr>
<td>Deterding</td>
<td>25.38</td>
<td>46.59</td>
</tr>
</tbody>
</table>
7.1 CONCLUSION

Factor Analysis (EM_FA) based on expectation-maximization algorithm, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) were used as dimensionality reduction techniques. Local factor analysis (EM based) and Local Principal Component Analysis were implemented using Vector Quantization. The results obtained from experiments are summarised below.

(1) While comparing PCA, LDA and EM_FA for global dimensionality reduction on some standard machine learning databases, it is found that the recognition accuracy for LDA is higher than the other two in most of the cases. It is found that the recognition accuracy for LDA method is 98.67%, 51.97%, 85.28%, 97.73%, 80.11% and 94.50% for IRIS, PB, GLASS, WINE, BCW and Deterding datasets respectively.

(2) FA (EM based) and PCA were applied for local dimensionality reduction using vector quantization. The performances of VQPCA and VQ_EM_FA are compared for dimensionality reduction, and VQPCA is found performing better than VQ_EM_FA. When the number of partitions were 4, the recognition accuracy of VQPCA for IRIS, PB and Deterding datasets was 96%, 31.32% and 70.64% respectively. Whereas, in the case of VQ_EM_FA, the recognition accuracy was found to be 94.67%, 31.84% and 59.09% respectively.
The local dimensionality reduction algorithms, VQPCA and VQ_EM_FA, partition the input space using the VQ and build PCA and EM_FA models respectively in each local region. Global models, such as PCA, FA, can incur a large error when data has different structure in different regions of the input space. Building simple linear models in local regions of the input space can be more accurate. After comparing FA with VQ-EM_FA and PCA with VQPCA, it has been found that VQFA and VQPCA perform better than FA and PCA, respectively, as the number of partitions (levels) of the input space is increased.

### 7.2 FUTURE WORK

Modeling data with Gaussian distribution is one of the main statistical problems and factor analysis can be further explored in making robust models. Diagonal Covariance Modeling is extensively used in speech recognition[80][81]. Techniques may be developed using factor analysis which can capture correlations in the speech data with few parameters.

The number of factors required in a factor model is quite ambiguous. Determining the appropriate number of common factors in the factor analysis model is an area where the development of some heuristic algorithm can be explored.
Factor analysis can be used in developing techniques to improve the correlation model of feature vector elements in speech recognition systems. Generative Factor Analyzed Hidden Markov Models[71], Factor Analyzed Hidden Markov Models for Speech Recognition[67], Factor Analysis Invariant to Linear Transformation of Data[80], The EM Algorithm for Mixtures of Factor Analysers [79], etc are some of the emerging areas where research work can be done and some new algorithms may be explored.
References


REFERENCES


REFERENCES


References


(1) Discrete Karhunen-Loeve Expansion [39]

Let \( \mathbf{X} \) be an \( n \)-dimensional random vector. The \( \mathbf{X} \) can be represented without error by the summation of \( n \) linearly independent vectors as

\[
\mathbf{X} = \sum_{i=1}^{n} y_i \phi_i
\]

\[
= \mathbf{\Phi} \mathbf{Y}
\]

where

\[
\mathbf{\Phi} = \begin{bmatrix} \phi_1 & \ldots & \phi_n \end{bmatrix}
\]

\[
\mathbf{Y} = \begin{bmatrix} y_1 & \ldots & y_n \end{bmatrix}^T
\]

The matrix \( \mathbf{\Phi} \) is deterministic and is made up of \( n \) linearly independent column vectors. Thus,

\[
| \mathbf{\Phi} | \neq 0
\]

The columns of \( \mathbf{\Phi} \) span the \( n \)-dimensional space containing \( \mathbf{X} \) and are called the basis vectors. The columns of \( \mathbf{\Phi} \) form an orthonormal set, that is

\[
\phi_i^T \phi_j = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}
\]

If the orthonormality condition is satisfied, the components of \( \mathbf{Y} \) can be calculated by

\[
y_i = \phi_i^T \mathbf{X} \quad (i = 1, \ldots, n)
\]

Thus, \( \mathbf{Y} \) is an orthonormal transformation of the random vector \( \mathbf{X} \). Here \( \phi_i \) the \( i \)th feature or feature vector, and \( y_i \) the \( i \)th component of the sample in the feature (or mapped) space.

Now we select only \( m \) (<\( n \)) of \( \phi_i \)'s and we will use these to approximate \( \mathbf{X} \). This we can do by replacing those components of \( \mathbf{Y} \), which we do not calculate, with preselected constants and form the following approximation

\[
\hat{\mathbf{X}}(m) = \sum_{i=1}^{m} y_i \phi_i + \sum_{i=m+1}^{n} b_i \phi_i
\]

Thus by doing this, only first \( m \) \( y \)'s are calculated.
The resulting representation error will be
\[
\Delta X(m) = X - X^\wedge(m) \\
= X - \sum_{i=1}^{m} y_i \phi_i - \sum_{i=m+1}^{n} b_i \phi_i \\
= \sum_{i=m+1}^{n} (y_i - b_i) \phi_i
\]

Mean-square magnitude of $\Delta X$ can used as a criterion to measure the effectiveness of the subset of $m$ features.

\[
\varepsilon^{-2}(m) = E\left\|\Delta X(m)\right\|^2 \\
= E\left\{ \sum_{i=m+1}^{n} \sum_{j=m+1}^{n} (y_i - b_i)(y_j - b_j) \phi_i^T \phi_j \right\} \\
= \sum_{i=m+1}^{n} E\{ (y_i - b_i)^2 \}
\]

For every choice of basis vectors and constant terms, we obtain a value for $\varepsilon^{-2}(m)$. We want to make a choice which minimizes $\varepsilon^{-2}(m)$.

The optimum choice of for $b_i$ is obtained by minimizing $\varepsilon^{-2}(m)$ with respect to $b_i$ as

\[
\frac{\partial}{\partial b_i} E\{ (y_i - b_i)^2 \} = -2[E\{y_i\} - b_i]
\]

Now

\[-2[E\{y_i\} - b_i] = 0\]

Solving for $b_i$

\[b_i = E\{y_i\} = \phi_i^T E\{X\}\]

Thus, we can replace those $y_i$’s, which we do not measure, by their expected values.
The mean square error can be written as

\[ e^{-2}(m) = \sum_{i=m+1}^{n} E\{(y_i - b_i)^2\} \]

\[ = \sum_{i=m+1}^{n} E\{(y_i - E\{y_i\})^2\} \]

\[ = \sum_{i=m+1}^{n} \phi_i^T E[(X - E\{X\})(X - E\{X\})^T] \phi_i \]

\[ = \sum_{i=m+1}^{n} \phi_i^T \Sigma_x \phi_i \]

where \( \Sigma_x \) is the covariance matrix of \( X \). The optimum choice for the \( \phi_i \) is those which satisfy

\[ \Sigma_x \phi_i = \lambda_i \phi_i \]

that is, the eigenvectors of \( \Sigma_x \). Thus the minimum mean-square error becomes

\[ e^{-2}(m)_{opt} = \sum_{i=m+1}^{n} \lambda_i \]

The expansion of a random vector in the eigenvectors of the covariance matrix is called the discrete version of the Karhunen-Loeve (K-L) expansion.
(2) Linear Discriminant Analysis [39]

In LDA, within class \(S_w\) and between class \(S_b\) scatter matrices are used to formulate criterion of class separability.

**Within Class Scatter Matrix:** it shows the scatter of samples around their respective class expected vectors, and is expressed by

\[
S_w = \sum_{i=1}^{L} P_i E\{E\{X - M_i\}(X - M_i)^T|\omega_i\}
\]

\[
= \sum_{i=1}^{L} P_i \Sigma_i
\]

**Between Class Scatter Matrix:** it shows the scatter of the expected vectors around the mixture mean

\[
S_b = \sum_{i=1}^{L} P_i (M_i - M_0)(M_i - M_0)^T
\]

where \(M_0\) represents the expected vector of the mixture distribution and is given by

\[
M_0 = E\{X\}
\]

\[
= \sum_{i=1}^{L} P_i M_i
\]

For class separability, we convert these matrices into a number. This number should be larger when \(S_b\) scatter is larger or \(S_w\) scatter is smaller. The criterion we have used for class separability is

\[
J_1 = \text{tr}(S_w^{-1}S_b)
\]

A linear transformation from an \(n\)-dimensional \(X\) to an \(m\)-dimensional \(Y\) \((m<n)\) can be expressed as

\[
Y = A^T X
\]

where \(A\) is an \(n \times m\) rectangular matrix and the column vectors are linearly independent. These column vectors do not need to be orthonormal. Since \(S_w\) and \(S_b\) have the form of a covariance matrix, \(S_i\) and \(S_2\) in the \(Y\)-space can be calculated from \(S_1\) and \(S_2\) from the \(X\)-space by

\[
S_{iY} = A^T S_{iX} A \quad \text{for } i=1,2
\]

hence in feature extraction for classification, we find \(A\) which optimizes \(J_1\) in the \(Y\)-space.

**Optimization of \(J_1\)**

Let \(J_1(m)\) be the value of \(J_1\) in an \(m\)-dimensional \(Y\)-space. Then,
\[ J_1(m) = \text{tr}(S_2Y^{-1} S_1Y) \]
\[ = \text{tr}\{(A^T S_2X A)^{-1}(A^T S_1X A)\} \]

We will differentiate the above equation with respect to \( A \)
\[ \frac{\partial J_1(m)}{\partial A} = -2 S_2X A S_2Y^{-1} S_1Y S_2Y^{-1} + 2 S_1X A S_2Y^{-1} \]

the optimum \( A \) must satisfy
\[ (S_2X^{-1} S_1X) A = A (S_2Y^{-1} S_1Y) \]

Two matrices \( S_1Y \) and \( S_2Y \) can be simultaneously diagonalized to \( \Omega_m \) and \( I_m \) by a linear transformation \( Z = B^TY \) such that
\[ B^T S_1Y B = \Omega_m \quad \text{and} \quad B^T S_2Y B = I_m \]

where \( B \) is an \( m \times m \) nonsingular square matrix and \( B^{-1} \) exists.

The criterion value is invariant under this non-singular transformation from \( Y \) to \( Z \).
\[ \text{tr}(S_2Z^{-1} S_1Z) = \text{tr}\{(B^T S_2Y B)^{-1}(B^T S_1Y B)\} \]
\[ = \text{tr}(B^{-1} S_2Y^{-1} B B^{-1} B^T S_1Y B) \]
\[ = \text{tr}(B S_2Y^{-1} S_1Y B B^{-1}) \]
\[ = \text{tr}(S_2Y^{-1} S_1Y) \]

Thus we can write
\[ (S_2X^{-1} S_1X) A = A (B \Omega_m B^{-1}) \]
or
\[ (S_2X^{-1} S_1X) (AB) = (AB) \Omega_m \]

This equation shows that the components of \( \Omega_m \) and the columns vectors of \( (AB) \) are the \( m \) eigenvalues and eigenvectors of \( S_2X^{-1} S_1X \).

Since the trace of a matrix is the summation of the eigenvalues,
\[ J_1(n) = \text{tr}(S_2X^{-1} S_1X) = \lambda_1 + \lambda_2 + \cdots + \lambda_n \]
\[ J_1(m) = \text{tr}(S_2Y^{-1} S_1Y) = \mu_1 + \mu_2 + \cdots + \mu_m \]

where \( \lambda_i \)'s and \( \mu_i \)'s are the eigenvalues of \( S_2X^{-1} S_1X \) and \( S_2Y^{-1} S_1Y \) respectively. We can maximize \( J_1(m) \) by selecting the largest \( m \) eigenvalues. The corresponding \( m \) eigenvectors form the transformation matrix. By projecting \( X \) onto the \( m \) eigenvectors of \( S_2X^{-1} S_1X \), we can form an \( m \) dimensional subspace which is spanned by these \( m \) eigenvectors. \( J_1(m) \) is the summation of the corresponding \( m \) eigenvalues.

The values of \( J_1(m) \) is attached to the subspace and is invariant regardless of the selection of the coordinate system in the subspace. Thus, it is appropriate to represent the subspace by the set of \( m \) eigenvectors of \( S_2X^{-1} S_1X \).

**Two-class Problems**

Let \( J_1 = \text{tr} (S_w^{-1} S_b) \) be the criterion for the two-class problems. For two-class
problems, $S_b$ becomes
\[ S_b = P_1 (M_1 - M_0)(M_1 - M_0)^T + P_2 (M_2 - M_0)(M_2 - M_0)^T \]
\[ = P_1 P_2 (M_2 - M_1)(M_2 - M_1)^T \]
where $P_1 M_1 + P_2 M_2 = M_0$.

Since $S_b$ is composed of one vector $(M_2 - M_1)$, the rank of $S_b$ is one. Since $S_w$ is the averaged covariance matrix, it generally has full rank and $S_w^{-1}$ exists. Therefore, the rank of $S_w^{-1}S_b$ is also one.

\[ \lambda_1 \neq 0 \text{ and } \lambda_2 = \ldots = \lambda_n = 0 \]

The $\text{tr}(S_w^{-1}S_b)$ is the summation of these eigenvalues,
\[ \lambda_1 = \text{tr}(S_w^{-1}S_b) = P_1 P_2 (M_2 - M_1)^T S_w^{-1} (M_2 - M_1) \]
the corresponding eigenvector is
\[ \phi_1 = \frac{S_w^{-1}(M_2 - M_1)}{\left\| S_w^{-1}(M_2 - M_1) \right\|} \]
where the denominator is a constant selected to satisfy $\| \phi_1 \| = 1$.

For two-class problems, only one feature is needed and others do not contribute to the value of $J_1$. The mapping function is
\[ y_1 = \phi_1^T X = c (M_2 - M_1)^T S_w^{-1} X \]
where $c$ is a constant.

**L-class Problems**

Two-class problems can be extended to $L$-class problems, still using $J_1 = \text{tr}(S_w^{-1}S_b)$. Since the $M_i$'s are related by eq., only $(L-1)$ of them are linearly independent. Therefore, $S_b$ has a rank of $(L-1)$, and subsequently the rank of $S_w^{-1}S_b$ is $(L-1)$. This means that $(L-1)$ eigenvalues of $S_w^{-1}S_b$ are nonzero and the others are zero. Thus, without losing the criterion value, we can map $X$ onto the $(L-1)$ dimensional subspace spanned by the $(L-1)$ eigenvectors corresponding to these nonzero eigenvalues.

In order to classify $L$ distributions, we need at least $(L-1)$ features. The optimization of $\text{tr}(S_w^{-1}S_b)$ also produces $(L-1)$ features without losing classifiability, as long as the classifiability is measured by $\text{tr}(S_w^{-1}S_b)$. 
(3) Expectation-Maximization Algorithm for Factor Analysis [74]

The expected log likelihood for factor analysis

\[
Q = E \left[ \log \prod_i (2\pi)^{p/2} |\Psi|^{-1/2} \exp \left\{ -\frac{1}{2} [X_i - \Lambda Z]'\Psi^{-1}[X_i - \Lambda Z] \right\} \right]
\]

\[
= c - \frac{n}{2} \log |\Psi| - \sum_i E \left[ \frac{1}{2} X_i'\Psi^{-1}X_i - X_i'\Psi^{-1}\Lambda Z + \frac{1}{2} Z'\Lambda'\Psi^{-1}\Lambda Z \right]
\]

\[
= c - \frac{n}{2} \log |\Psi| - \sum_i E \left[ \frac{1}{2} X_i'\Psi^{-1}X_i - X_i'\Psi^{-1}\Lambda \ E[Z|X_i] + \frac{1}{2} \text{tr} [\Lambda'\Psi^{-1}\Lambda \ E[ZZ'|X_i]] \right]
\]

where \( c \) is a constant, independent of the parameters, and \( \text{tr} \) is the trace operator.

To re-estimate the factor loading matrix, we differentiate \( Q \) with respect to \( \hat{\Lambda} \),

\[
\frac{\partial Q}{\partial \Lambda} = - \sum_i \Psi^{-1}X_i E[Z|X_i]' + \sum_i \Psi^{-1}\Lambda^{\text{new}} \ E[ZZ'|X_i] = 0
\]

\[
\Lambda^{\text{new}} \left( \sum_i E[ZZ'|X_i]' \right) = \sum_i X_i E[Z|X_i]'
\]

\[
\Lambda^{\text{new}} = \left( \sum_{i=1}^n X_i E[Z|X_i]' \right) \left( \sum_{i=1}^n E[ZZ'|X_i] \right)^{-1}
\]

We can estimate the matrix \( \phi \) through its inverse, setting

\[
\frac{\partial Q}{\partial \Psi^{-1}} = \frac{n}{2} \Psi^{\text{new}} - \sum_i \left( \frac{1}{2} X_i X_i' - \Lambda^{\text{new}} \ E[Z|X_i]' X_i + \frac{1}{2} \Lambda^{\text{new}} \ E[ZZ'|X_i] \Lambda^{\text{new}'} \right) = 0
\]

we can get

\[
\frac{n}{2} \Psi^{\text{new}} = \sum_i \frac{1}{2} X_i X_i' - \frac{1}{2} \Lambda^{\text{new}} \ E[Z|X_i]' X_i
\]

\[
\Psi^{\text{new}} = \frac{1}{n} \text{diag} \left\{ \sum_{i=1}^n X_i X_i' - \Lambda^{\text{new}} \ E[Z|X_i]' X_i \right\}
\]

where the \( \text{diag} \) operator sets all the off-diagonal elements of a matrix to zero.
(3.1) Proof of the Matrix Inversion Lemma

\[
( A + X B X^T )^{-1} = A^{-1} - A^{-1} X ( B^{-1} + X^T A^{-1} X )^{-1} X^T A^{-1}
\]

To prove the above lemma, we can show that

\[
( A^{-1} - A^{-1} X ( B^{-1} + X^T A^{-1} X )^{-1} X^T A^{-1} ) ( A + X B X^T ) = I
\]

L.H.S

\[
= I + A^{-1} X B X^T - A^{-1} X ( B^{-1} + X^T A^{-1} X )^{-1} X^T - A^{-1} X ( B^{-1} + X^T A^{-1} X )^{-1} X^T A^{-1} X B X^T
\]

\[
= I + A^{-1} X ( B X^T - ( B^{-1} + X^T A^{-1} X )^{-1} X^T - ( B^{-1} + X^T A^{-1} X )^{-1} X^T A^{-1} X B X^T )
\]

\[
= I + A^{-1} X ( B X^T - ( B^{-1} + X^T A^{-1} X )^{-1} B X^T - ( B^{-1} + X^T A^{-1} X )^{-1} X^T A^{-1} X B X^T )
\]

\[
= I + A^{-1} X ( B X^T - B X^T )
\]

\[
= I
\]
(1) Intermediate Results (PB dataset) - PCA

=================================================================
pbtrain (10 classes, 4 dimensions, 760 vectors)
pbtest (10 classes, 4 dimensions, 760 vectors)
=================================================================

D=4

<table>
<thead>
<tr>
<th>47</th>
<th>28</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>61.84</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>15</td>
<td>18</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19.74</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>14</td>
<td>25</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6 18.42</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>11</td>
<td>41</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>53.95</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>39</td>
<td>4</td>
<td>2</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>6 51.32</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>50</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>65.79</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>50</td>
<td>3</td>
<td>13</td>
<td>2</td>
<td>65.79</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>5</td>
<td>25</td>
<td>24</td>
<td>5</td>
<td>32.89</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>15</td>
<td>11</td>
<td>45</td>
<td>2</td>
<td>59.21</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>68 89.47</td>
</tr>
</tbody>
</table>

Final Score = 394 / 760
Final Score (percent) = 51.8421

D=3

<table>
<thead>
<tr>
<th>47</th>
<th>28</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>61.84</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>15</td>
<td>19</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19.74</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>14</td>
<td>26</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5 18.42</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>11</td>
<td>41</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>53.95</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>39</td>
<td>4</td>
<td>2</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>65.79</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>50</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>65.79</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>5</td>
<td>25</td>
<td>24</td>
<td>5</td>
<td>32.89</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>15</td>
<td>11</td>
<td>45</td>
<td>2</td>
<td>57.89</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>67 88.16</td>
</tr>
</tbody>
</table>

Final Score = 388 / 760
Final Score (percent) = 51.0526

D=2

<table>
<thead>
<tr>
<th>47</th>
<th>15</th>
<th>13</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>61.84</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>11</td>
<td>5</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14.47</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>5</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.58</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>15</td>
<td>29</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>38.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>32</td>
<td>5</td>
<td>0</td>
<td>14</td>
<td>13</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>12</td>
<td>27</td>
<td>3</td>
<td>9</td>
<td>20</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>28</td>
<td>1</td>
<td>30</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>18</td>
<td>11</td>
<td>4</td>
<td>9</td>
<td>27</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>17</td>
<td>4</td>
<td>38</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>66</td>
<td></td>
</tr>
</tbody>
</table>

**Final Score = 292 / 760**  
**Final Score(percent) = 38.4211**

|    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 46 | 17 | 12 | 1  | 0  | 0  | 0  | 0  | 0  | 0  |
| 29 | 11 | 3  | 30 | 3  | 0  | 0  | 0  | 0  | 0  |
| 17 | 19 | 6  | 17 | 17 | 0  | 0  | 0  | 0  | 0  |
| 8  | 9  | 13 | 19 | 26 | 1  | 0  | 0  | 0  | 0  |
| 0  | 2  | 2  | 12 | 23 | 9  | 12 | 6  | 6  | 4  |
| 0  | 1  | 1  | 3  | 23 | 12 | 17 | 9  | 4  | 6  |
| 0  | 0  | 0  | 4  | 7  | 6  | 42 | 2  | 10 | 5  |
| 0  | 0  | 1  | 11 | 9  | 9  | 33 | 5  | 2  | 6  |
| 0  | 0  | 0  | 4  | 9  | 5  | 45 | 6  | 4  | 3  |
| 0  | 0  | 0  | 4  | 12 | 7  | 37 | 6  | 1  | 9  |

**Final Score = 177 / 760**  
**Final Score(percent) = 23.2895**

(2) Intermediate Results (PB dataset) - LDA

-----------------------------------------------

pbtrain (10 classes, 4 dimensions, 760 vectors)  
pbtest (10 classes, 4 dimensions, 760 vectors)

-----------------------------------------------

|    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 47 | 28 | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 29 | 15 | 19 | 12 | 0  | 0  | 0  | 0  | 0  | 1  |
| 15 | 15 | 14 | 26 | 1  | 0  | 0  | 0  | 0  | 5  |
| 4  | 2  | 11 | 41 | 8  | 0  | 0  | 0  | 0  | 10 |
| 0  | 0  | 1  | 6  | 40 | 3  | 2  | 18 | 0  | 6  |
| 0  | 0  | 2  | 1  | 50 | 10 | 10 | 0  | 3  | 65 |
| 0  | 0  | 0  | 0  | 2  | 6  | 48 | 3  | 16 | 1  |
| 0  | 0  | 0  | 0  | 17 | 0  | 5  | 25 | 24 | 5  |
| 0  | 0  | 1  | 0  | 3  | 0  | 14 | 10 | 47 | 1  |
| 0  | 0  | 1  | 7  | 0  | 0  | 0  | 0  | 68 | 9  |

**Final Score = 395 / 760**  
**Final Score(percent) = 51.9737**  
**D=4**

|    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 47 | 28 | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 29 | 15 | 19 | 12 | 0  | 0  | 0  | 0  | 0  | 1  |
| 15 | 15 | 14 | 26 | 1  | 0  | 0  | 0  | 0  | 5  |
| 4  | 2  | 11 | 41 | 8  | 0  | 0  | 0  | 0  | 10 |
| 0  | 0  | 1  | 6  | 40 | 3  | 2  | 18 | 0  | 6  |
| 0  | 0  | 2  | 1  | 50 | 10 | 10 | 0  | 3  | 65 |
| 0  | 0  | 0  | 0  | 2  | 6  | 48 | 3  | 16 | 1  |
| 0  | 0  | 0  | 0  | 17 | 0  | 5  | 25 | 24 | 5  |
| 0  | 0  | 1  | 0  | 3  | 0  | 14 | 10 | 47 | 1  |
| 0  | 0  | 1  | 7  | 0  | 0  | 0  | 0  | 68 | 9  |

**Final Score = 395 / 760**  
**Final Score(percent) = 51.9737**  
**D=4**
Final Score =  393  /  760
Final Score(percent) =  51.7105  D=3

D=2

46  29  1  0  0  0  0  0  0  0  60.53
29  15  23  7  0  0  0  0  0  2  19.74
14  14  17  15  12  0  0  0  0  4  22.37
3  2  11  35  25  0  0  0  0  0  46.05
0  0  0  12  36  1  14  12  1  0  47.37
0  0  0  5  5  42  23  1  0  0  55.26
0  0  0  1  7  8  51  4  5  0  67.11
0  0  0  10  8  0  0  15  34  9  19.74
0  0  0  4  3  0  0  9  51  9  67.11
0  0  3  4  5  0  0  4  32  28  36.84

Final Score =  336  /  760
Final Score(percent) =  44.2105  D=2

D=1

46  15  14  1  0  0  0  0  0  0  60.53
28  12  5  30  1  0  0  0  0  0  15.79
16  20  6  23  11  0  0  0  0  0  7.89
7  9  13  22  25  0  0  0  0  0  28.95
0  0  2  10  26  5  4  8  12  9  34.21
0  0  0  5  17  7  11  16  10  10  9.21
0  0  0  4  1  7  45  6  12  1  59.21
0  0  0  10  11  3  27  10  8  7  13.16
0  0  0  4  6  1  43  12  8  2  10.53
0  0  1  8  18  5  17  11  9  7  9.21

Final Score =  189  /  760
Final Score(percent) =  24.8684  D=1

(3) Intermediate Results (PB dataset) - EM_FA

====================================================================
pbtrainC2  (10 classes, 4 dimensions, 760 vectors)
pbtestC2   (10 classes, 4 dimensions, 760 vectors)
====================================================================

D=4

47  28  1  0  0  0  0  0  0  0  61.84
29  14  17  15  0  0  0  0  0  1  18.42
15  14  15  26  1  0  0  0  0  5  19.74
4  2  11  42  8  0  0  0  0  0  55.26
0  0  1  5  42  2  2  18  0  0  55.26
0  0  0  2  1  49  8  12  0  4  64.47
0  0  0  0  2  45  4  20  3  39.21
0  0  0  0  17  0  2  23  27  7  30.26
0  0  1  0  3  0  10  10  50  2  65.79
0  0  1  7  1  0  0  0  0  67  88.16
**Final Score** = 394 / 760  
**Final Score D=4 (percent)** = 51.8421

<table>
<thead>
<tr>
<th>D=3</th>
<th>47</th>
<th>28</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>61.84</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>29</td>
<td>14</td>
<td>18</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>18.42</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>14</td>
<td>15</td>
<td>26</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>19.74</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>11</td>
<td>42</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>55.26</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>42</td>
<td>2</td>
<td>2</td>
<td>18</td>
<td>0</td>
<td>6</td>
<td>55.26</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>48</td>
<td>10</td>
<td>12</td>
<td>0</td>
<td>3</td>
<td>63.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>44</td>
<td>5</td>
<td>19</td>
<td>3</td>
<td>57.89</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>2</td>
<td>24</td>
<td>27</td>
<td>6</td>
<td>31.58</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>11</td>
<td>48</td>
<td>2</td>
<td>63.16</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>67</td>
</tr>
</tbody>
</table>

**Final Score** = 391 / 760  
**Final Score D=3 (percent)** = 51.4474

<table>
<thead>
<tr>
<th>D=2</th>
<th>47</th>
<th>24</th>
<th>5</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>61.84</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>29</td>
<td>12</td>
<td>12</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>15.79</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>18</td>
<td>9</td>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>11.84</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>8</td>
<td>39.47</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>27</td>
<td>5</td>
<td>0</td>
<td>15</td>
<td>16</td>
<td>4</td>
<td>35.53</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>37</td>
<td>6</td>
<td>6</td>
<td>20</td>
<td>1</td>
<td>48.68</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>28</td>
<td>1</td>
<td>29</td>
<td>0</td>
<td>36.84</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>16</td>
<td>4</td>
<td>0</td>
<td>12</td>
<td>29</td>
<td>8</td>
<td>15.79</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>14</td>
<td>8</td>
<td>4</td>
<td>35</td>
<td>7</td>
<td>46.05</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>66</td>
</tr>
</tbody>
</table>

**Final Score** = 303 / 760  
**Final Score D=2 (percent)** = 39.8684

<table>
<thead>
<tr>
<th>D=1</th>
<th>47</th>
<th>16</th>
<th>12</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>61.84</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>29</td>
<td>11</td>
<td>3</td>
<td>24</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14.47</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>16</td>
<td>8</td>
<td>12</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10.53</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>16</td>
<td>19</td>
<td>9</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>21.05</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>12</td>
<td>18</td>
<td>8</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>14</td>
<td>23.68</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>19</td>
<td>11</td>
<td>3</td>
<td>12</td>
<td>2</td>
<td>18</td>
<td>14.47</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>35</td>
<td>13.16</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>6</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td>32</td>
<td>11.84</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>43</td>
<td>3.95</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>12</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>47</td>
<td>61.84</td>
<td></td>
</tr>
</tbody>
</table>

**Final Score** = 180 / 760  
**Final Score D=1 (percent)** = 23.6842

◆◆◆