THESIS

MIXTURE OF FACTOR MODELS FOR JOINT DIMENSIONALITY REDUCTION AND CLASSIFICATION

Submitted by

Puoya Tabaghi

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2016

Master's Committee:

Advisor: Mahmood R. Azimi-Sadjadi
Co-Advisor: Louis L. Scharf

Ali Pezeshki
Michael Kirby

ABSTRACT

MIXTURE OF FACTOR MODELS FOR JOINT DIMENSIONALITY REDUCTION AND CLASSIFICATION

In many areas such as machine learning, pattern recognition, information retrieval, and data mining one is interested in extracting a low-dimensional data that is truly representative of the properties of the original high dimensional data. For example, one application could be extracting representative low-dimensional features of underwater objects from sonar imagery suitable for detection and classification. This is a difficult problem due to various factors such as variations in the operating and environmental conditions, presence of spatially varying clutter, and variations in object shapes, compositions, and orientation.

The goal of this work is to develop a novel probabilistic method using a mixture of factor models for simultaneous nonlinear dimensionality reduction and classification. The framework used here is inspired by the work in [1] which uses a mixture of local PCA projections leading to an unsupervised nonlinear dimensionality reduction algorithm. In contrast, the proposed method provides a supervised probabilistic approach suitable for analyzing labeled high-dimensional data with complex structures by exploiting a set of low-dimensional latent variables which are both discriminative and generative. With the aid of these low-dimensional latent variables, a mixture of linear models is introduced to represent the high-dimensional data. An optimum linear classifier is then built in the latent variable-domain to separate the support of the latent variable associated with each class. Introducing these hidden variables allow us to derive the joint probability density function of the data and class label, reduce data dimension and perform clustering, classification and parameter estimation. This probabilistic approach provides a mechanism to traverse between the input space and latent (feature) space and vice versa as well as cluster and classify data.

A supervised training based on the Expectation-Maximization (EM) and steepest descent algorithms is then introduced to derive the ML estimates of the unknown parameters. It is shown that parameters associated with dimensionality reduction can be estimated using the EM algorithm whereas those of the classifier are estimated using the steepest descent method. The introduction of latent variables not only helps to represent the pdf of data and reduce the dimension of them but also in parameter estimation using EM algorithm which is used to find ML estimates of the parameters when the available data is incomplete.

A comprehensive study is carried out to assess the performance of the proposed method using two different data sets. The first data set consists of Synthetic Aperture Sonar (SAS) images of model-generated underwater objects superimposed on background clutter. These images correspond to two different object types namely Cylinder (mine-like) and Block (non-mine-like). The signatures of each object are synthetically generated and are placed at various aspect angles from 1 to 180 degrees for each object type. The goal of our classifier is to assign non-target versus target labels to these image snippets. The other data set consists of two sets of facial images of different individuals. Each image set contains 2 series of 93 images of the same person at different poses. The goal of the classifier for this case is to idnetify each individual correctly. The dimensionality reduction performance of the proposed method is compared to two relevant dimensionality reduction methods, namely Probabilistic PCA [2] and Mixture of Probabilistic PCA (MPPCA) [1] while its classification performance is benchmarked against a Support Vector Machine (SVM). The results on both data sets indicate promising dimensionality reduction and reconstruction capabilities compared to PPCA/MPPCA methods. On the other hand, classification performance is competitive with SVM when the data is linearly separable.

ACKNOWLEDGEMENTS

I would first like to thank my adviser, Prof. Mahmood R. Azimi-Sadjadi, for his invaluable support and guidance throughout the course of this research. I would also like to thank Prof. Louis Scharf for his useful suggestions throughout the course of this research. They have taught me the value of developing the intellectual reasoning needed to conduct thorough and well-developed research. Their guidance and time is greatly appreciated throughout the course of my graduate education.

I would like to thank my committee members, Prof. Michael Kirby and Prof. Ali Pezeshki, for their time and assistance.

I would like to thank the Office of Naval Research (ONR) and the Naval Surface Warfare Center (NSWC) - Panama City, Florida for providing the funding and the data used in this project. This project was funded by the ONR-3210E under contract number N00014-14-1-011O.

I would like to express my appreciation for my colleagues at the Colorado State University, Signal and Image Processing Laboratory for providing a fun and supportive work environment. In particular, thanks to Jack Hall, Vladimir Yaremenko, Pooria Pakrooh and Jarrod Zacher.

Finally, I will forever be grateful to my family for their love, support and guidance throughout my life.

DEDICATION

To my parents, Javad and Mehri, and my sister, Shiva.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement and Motivations

The *curse of dimensionality* expression was coined by Bellman in 1961 when considering problems in dynamic optimization [3]. It refers to the fact that many algorithms become intractable when the input data is high dimensional [4]. For instance, when designing a classifier, if the dimension of input data is very large, the number of unknown parameters to be identified will be exceedingly large and hence an enormous training set will be needed to avoid overfitting. Therefore, it is necessary to first reduce the dimension of the input data to a manageable size while keeping as much of the original information as possible before applying it to the classifier. In general, dimensionality reduction can bring an improved understanding of the data apart from a computational advantage. Dimensionality reduction can also be viewed as a feature extraction or mapping for representing the data in a different coordinate system. The problem of dimensionality reduction can be formally defined as follows. Suppose we have a data set $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$ where $\mathbf{x}_n \in \mathbb{R}^D$. The fundamental assumption that justifies dimensionality reduction is that the data actually lies, at least approximately, on a low dimensional manifold of dimension $d \ll D$ that needs to be discovered. The goal of dimensionality reduction is to find an approximate representation of that manifold which will allow us to obtain a low-dimensional, compact representation of the data.

On the other hand, the problem of *classification* has been widely studied in decision-making systems, pattern recognition, data mining, and information retrieval communities. The problem of supervised classification is defined as follows. Given a set of training data samples $\mathcal{D} = \{\mathbf{x}_n, z_n\}_{n=1}^N$ where $\mathbf{x}_n \in \mathbb{R}^D$ and $z_n \in \{1, ..., K\}$ represents the associated class label, we would like to construct a classification system that associates a new data sample, $\mathbf{x}_{new}$ to one of the K possible classes. This is done via a proper training procedure which

converges to a set of solutions for the unknown parameters of the classifier. In the hard version of the classification problem, a particular label is explicitly assigned to the sample, whereas in the soft version of the classification problem, a probability value is assigned to the test sample. The classification problem assumes categorical values for the labels, though it is also possible to use continuous values as labels as well. The latter is referred to as the regression modeling problem [5].

Now, consider joint classification and dimensionality reduction problems. One might be interested in extracting low-dimensional features that are truly representative of the properties of the original high-dimensional data and also determine its label at the same time. When designing feature extraction and classification algorithms separately, the questions are: (1) How well the low-dimensional features will do in classification? (2) Could there be any connections between appropriate classification features with the resultant reduced dimension features? Finding answers to these questions motivated us to develop a supervised probabilistic approach for analyzing labeled high dimensional data with complex structures by exploiting a set of low dimensional latent variables (features). Thus, the method proposed in this work, provides a novel probabilistic method for simultaneous dimensionality reduction and classification.

1.2 Survey of Previous Works

The problem of dimensionality reduction, extracting low dimensional structure from high dimensional data, arises often in machine learning and statistical pattern recognition. High dimensional data takes many different forms: from digital image libraries to gene expression microarrays, from neuronal population activities to financial time series. We can categorize dimensionality reduction methods as linear or nonlinear, probabilistic or deterministic, etc. This section begins with a review of classical linear methods and then it focuses on popular nonlinear dimensionality reduction and manifold learning methods. Finally, the remainder of this section is devoted to probabilistic methods.

1.2.1 Linear Methods

If the data, labeled or unlabeled, is mainly confined to a low dimensional subspace, then simple linear methods can be used to discover the subspace. Among the most popular linear dimensionality reduction algorithms are Principal Component Analysis (PCA) [6, 7], Linear Discriminant Analysis (LDA) [8,36], and Factor Analysis [9]. *Principal Component Analysis* (PCA) has proven to be an exceedingly popular technique for dimensionality reduction and is discussed at length in most texts on multivariate analysis. Its many application areas include data reduction, image analysis, visualization, pattern recognition, regression and time series prediction. PCA retains maximal variance in the projected space subject to orthonormality of the mapping matrix. The main property of PCA is that the projection onto the principal subspace minimizes the squared reconstruction error. In this sense, PCA yields the best generative low dimensional features among orthogonal linear transformations, but it doesn't guarantee any discrimination for labeled data in the feature space. LDA, on the other hand, seeks to find a set of coordinate axes that carry the most discriminative signal components that are useful for classification though they are not suitable for reconstruction purposes. *Factor Analysis* is another linear (and probabilistic) dimensionality reduction method which estimates a set of low dimensional latent variables (or features) assuming a linear mapping from latent space to original data space and normal distributions for both latent variables and noise. The main drawback with these classical dimensionality reduction methods is that the low-dimensional subspace provides either a generative or discriminative representation of the original data, but not both.

1.2.2 Nonlinear Methods

Dimensionality reduction methods that capture the inherent nonlinear properties of the data in a lower dimensional manifold have attracted considerable attentions in recent years. Graph-based methods [10] have recently emerged as a powerful tool for analyzing high dimensional data that has been sampled from a low dimensional manifold. To analyze data that

lies on a low dimensional manifold, matrices are constructed from sparse weighted graphs whose vertices represent input patterns and whose edges indicate neighborhood relations. The resulting graph can be viewed as a discretized approximation of the manifold sampled by the input patterns. From these graphs, one can then construct matrices whose spectral decompositions reveal the low dimensional structure of the manifold. Spectral methods are able to reveal low dimensional structure in high dimensional data from the top or bottom eigenvectors of specially constructed matrices. In what follows, we mention four broadly representative graph-based spectral algorithms for manifold learning: Isometric Feature Mapping (ISOMAP) [11], Maximum Variance Unfolding (MVU) or semi-definite embedding [12], Locally Linear Embedding (LLE) [13], and Laplacian Eigenmaps [14].

The general idea principle behind ISOMAP is to use geodesic distances (not Euclidean distance which obscures the intrinsic manifold structure) on a graph together with the classical Multidimensional Scaling (MDS) [15]. Unlike ISOMAP which is a global approach, i.e. preserves geometry at all scales, LLE and Laplacian Eigenmaps are local approaches in that they attempt to only preserve the local geometry of the data by mapping nearby points on the manifold to nearby points in the low-dimensional representation. Similar to ISOMAP and LLE, MVU also belongs to the class of *spectral embedding*, however, it exploits different geometrical properties. ISOMAP is based upon geodesic distances, LLE on the coefficients of local linear reconstructions, and Laplacian eigenmaps on the discrete graph Laplacian, whereas MVU is based on estimating and preserving local distances and angles. In spite of their similarities, in [12] it was shown that for cases where the sampled manifold is not isometric to a convex subset of Euclidean space ISOMAP produces totally different results than that produced by MVU. Additionally, these methods exhibit inability to deal with out-of-sample extension for non-isometric manifolds. This implies that they fail to provide a feature mapping to map new data points that are not included in the original training set. Additionally and more importantly in forming the low dimensional feature space they don't account for the class membership of the data samples as they are inherently unsupervised.

1.2.3 Probabilistic Methods

One of the central problems in pattern recognition and machine learning is that of density estimation, i.e the construction of a model of a probability distribution given a finite sample of data drawn from that distribution. A powerful approach to probabilistic modeling involves supplementing a set of observed variables with additional latent, or hidden, variables. By defining a joint distribution over visible and latent variables, the corresponding distribution of the observed variables is then obtained by marginalization. This allows relatively complex distributions to be expressed in terms of more tractable joint distributions over the expanded variable space. On the other hand, estimating continous latent variables could amount to a feature extraction algorithm [7,16]. In what follows, we mention some broadly representative probabilistic algorithms, namely Probabilistic PCA [2], Mixture of Probabilistic PCA [1] and Generative Topographic Mapping [17].

1. *Probabilistic PCA* (PPCA): One limiting disadvantage of principal components analysis (PCA) is the absence of an associated probability density model for data. It is shown [2] how a particular form of linear latent variable model can be used to provide a probabilistic formulation of the standard PCA. This algorithm emerges from a Maximum Likelihood (ML) framework simply by assuming an isotropic noise distribution in a conventional Factor Analysis model. More specifically, there exists a unique ML estimate of the mapping matrix which is closely related to the d principal components of the data. It may also be shown that the ML estimator for noise variance is the average variance "lost" per discarded dimensions. The MMSE estimate of low-dimensional features is the posterior mean of latent variables which is closely related to features extracted by standard PCA.

   Deriving PCA from the perspective of density estimation would offer a number of important advantages. First, the corresponding likelihood would permit comparison with other density estimation techniques and facilitate statistical testing. Second,

Bayesian inference methods could be applied by combining the likelihood with a prior [18]. Third, the value of the probability density function could be used as a measure of the "degree of novelty" of a new data point.

2. *Mixture of Probabilistic PCA* (MPPCA) is derived extending the probabilistic model of PCA to a mixtures of models, [1], in an effort to retain a greater proportion of the variance using fewer components. Introducing the latent variables and using a mixture of linear models in the original data domain allows one to derive the probability density of the data. This formulation would permit all of the model parameters to be determined from ML, where both the appropriate partitioning of the data and the determination of the respective principal axes occur automatically as the likelihood is maximized. For the mixture of PPCA model, data points are assigned to the mixture components according to the posterior distribution associated with mixture components. In order to perform clustering for a data sample, a MAP estimate of mixture component is used [1]. The estimated low-dimensional feature in MPPCA model can also be generated using the postertior mean of the latent variable given the data sample and estimated model index.

   Similar to PPCA, in order to achieve least squares estimate of data sample, a reconstruction rule is written in terms of the projection matrix onto subspace associated with the estimated linear model. Implementation of this method was shown [1] to produce good results in image compression and handwritten digit recognition applications, though the reconstruction error attained is in general larger than that attained by a non-probabilistic mixture of PCAs trained to minimize the reconstruction error. This methods ignores class labels of data samples due to its unsupervised nature.

3. *Generative Topographic Mapping* (GTM) is an alternative nonlinear data modeling to MPPCA. A form of non-linear latent variable model called the Generative Topographic Mapping is introduced in [17] for which ML estimate of the model parameters can be determined using the EM algorithm

   A specific form for latent variables' prior is considered by a sum of delta functions centered on the specific nodes of a regular grid in latent space. Each node is then mapped to a corresponding point in data space, which forms the center of a Gaussian density function. It is easily shown that the data distribution is a constrained Gaussian mixture model [17] since the centers of the Gaussians cannot move independently but are related through the nonlinear mapping function. One application of GTM is data visualization, in which Bayes' theorem is used to invert the transformation from latent space to data space.

   GTM provides a principled alternative to the widely used Self-Organizing Map (SOM) of Kohonen [19], while overcoming most of its limitations. The most significant difference between the GTM and SOM algorithm is that GTM defines an explicit probability density given by the mixture distribution. As a consequence there is a well-defined objective function given by the log likelihood, and convergence to a local maximum of the objective function is guaranteed by using the EM algorithm. For the SOM algorithm, however, there is no probability density and no well-defined objective function that is being optimized by the training process. A further limitation of the SOM is that the conditions under which self-organization of the SOM occurs have not been quantified, and so it is necessary to confirm empirically that the trained model does indeed have the desired spatial ordering. In contrast, the neighborhood-preserving nature of the GTM mapping is an automatic consequence of the choice of a continuous mapping function.

1.3 Proposed Method

In this thesis, a novel probabilistic method using a *Mixture of Factor Models* (MFM) is proposed for simultaneous nonlinear dimensionality reduction and classification. The framework used here is inspired by the work in [1] which uses a mixture of local PCA projections leading to an unsupervised nonlinear dimensionality reduction algorithm. In contrast, this proposed method provides a supervised probabilistic approach for analyzing labeled high dimensional data with complex structures by exploiting a set of low dimensional latent variables. With the aid of these low-dimensional latent variables, a mixture of linear models is introduced to represent the high-dimensional data. An optimum linear classifier is then built in the latent variable-domain to separate the support of the latent variable associated with each class. Introducing these hidden variables, discrete and continuous, allows us to derive relatively complex joint distribution of data and labels to be expressed in terms of more tractable joint distributions. The ultimate goal is to reduce data dimension, perform clustering, classification, and parameter estimation. This probabilistic approach provides a mechanism to traverse between the input space and latent (feature) space and vice versa as well as to cluster and classify data.

A supervised training based on the EM and steepest descent algorithms is introduced to derive the ML estimates of the unknown parameters. It is shown that parameters associated with dimensionality reduction can be estimated using the EM algorithm whereas those of the classification system are estimated using a steepest descent algorithm. The introduction of latent variables not only helped representing the pdf of data and reducing the dimension of them but also in parameter estimation using EM algorithm which is used to find ML estimates of the parameters when the available data is incomplete. To define the explicit distribution for data in each class we derived new results for squashed multivariate normal distribution. It encompasses general cases of multivariate truncated normal and skew-normal distributions. Finding their distributions also helped us to find analytic solutions for the EM algorithm in parameter estimation.

A comprehensive study is then carried out to assess the performance of this proposed method using two different data sets. The first data set consists of SAS images of model-generated underwater objects superimposed on background clutter. These images correspond to two different object types namely Cylinder (mine-like) and Block (non-mine-like). The signatures of each object are synthetically generated and are inserted into backgrounds at various aspect angles (w.r.t sonar platform) from 1 to 180 degrees for each object type. The goal of our classifier here is to assign target versus non-target labels to the image snippets. The other data set consists of two sets of facial images of different individuals. Each image set contains of two 2 sets of 93 images of the same person at different poses. The goal of the classifier is to idnetify each of two selected individuals correctly. The dimensionality reduction performance of the proposed method is compared to Probabilistic PCA [2] and MPPCA [1] while its classification performance is benchmarked against SVM [20].

1.4 Organization of the Thesis

This thesis is organized as follows: Chapter 2 gives a detailed review of classical subspace data modeling and dimensionality reduction algorithms that use latent variables such as Factor Analysis, Probabilistic PCA and MPPCA. Chapter 3 introduces the general framework of the proposed method. Dimensionality reduction, clustering, and reconstruction are discussed in this chapter. This chapter also develops classification system based on both actual data points and estimated latent features. In Chapter 4, ML estimate of the parameters using EM and steepest descent algorithms is discussed. This algorithm was implemented in Chapter 5 and its reconstruction and classification performance are compared against those of PPCA, MPPCA, SVM, respectively. Finally, Chapter 6 concludes the studies carried out in this research and discusses possible ideas for future work.

CHAPTER 2

SUBSPACE DATA MODELING AND DIMENSIONALITY REDUCTION

2.1 Introduction

Most of the popular probabilistic dimensionality reduction methods, [1, 2, 9, 21] make use of latent variable models since they provide a simple and powerful tool for data analysis, e.g., when the intrinsic dimensionality of the problem is smaller than the apparent one [21]. In a broad sense many probabilistic models commonly used in machine learning can be considered as latent variable models inasmuch as they include probability distributions for variables which are not direclty observed. For example, in mixture models [22] the variable which indexes the components is a latent variable while in hidden Markov models [23] the state sequence is unobserved. Continuous latent variable models [21] are, on the other hand, especially suited for dimensionality reduction and missing data reconstruction.

In this chapter, we will concentrate exclusively on latent variable models where both the latent and the observed variables are continuous, going further than the famous linear model of factor analysis [24] [16]. One of the most popular techniques for dimensionality reduction, PCA has been recast [2] in the form of a particular kind of factor analysis. The association of a probability model with PCA offers the tempting prospect of being able to model complex data structures with a combination of local PCA models through the mechanism of a mixture of probabilistic principal component analyzers [1]. This chapter does not include any of the more recent latent variable models such as Generative Topographic Mapping (GTM) [17], Independent Component Analysis (ICA) [25] or Independent Factor Analysis (IFA) [26]. Instead, in this chapter we concentrate on factor analysis, probabilistic PCA (PPCA), and mixture of PPCA (MPPCA) algorithms.

The outline of this chapter is as follows. Section 2.2 gives a brief review of factor analysis and provides some background to the methods employed in the later sections of this chapter.

Section 2.3 illustrates how the principal axes of a set of observed data vectors may be determined through ML estimation of parameters in a latent variable model that is closely related to factor analysis. Section 2.4 provides nonlinear variants of PCA by mixing single PCA models to capture data complexity. Concluding remarks are then given in Section 2.5.

2.2 Factor Analysis

Factor Analysis [16] is probably the most common example of a statistical latent variable model which uses a linear mapping from latent space to data space. Typically, the latent variables are assumed to be independent and normal with unit variance, i.e. $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The latent variables $\mathbf{y}$ are often referred to as the *factors*. The mapping from latent space to data space, $\mathcal{M}$, is linear of form

$$\mathcal{M}(\mathbf{y}) = \mathbf{G}\mathbf{y} + \boldsymbol{\mu}, \tag{2.1}$$

where the columns of $\mathbf{G} \in \mathbb{R}^{D \times d}$ matrix are referred to as the *factor loadings*. Factor loading are assumed to be linearly independent, or $\text{rank}(\mathbf{G}) = d$. The parameter $\boldsymbol{\mu}$ permits the data model to have non-zero mean. Given the data model

$$\mathbf{x} = \mathcal{M}(\mathbf{y}) + \mathbf{n}. \tag{2.2}$$

where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \Psi)$ is white noise with diagonal covariance matrix Ψ, the distribution of $\mathbf{x}$ given $\mathbf{y}$ is normal centered at $\mathcal{M}(\mathbf{y})$, i.e.

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}(\mathcal{M}(\mathbf{y}), \Psi). \tag{2.3}$$

The D diagonal elements of Ψ are referred to as the *uniquenesses*. The distribution of data can be computed analytically by marginalizing the joint distribution of latent and observable (data) variables. Alternatively, using (2.1) and (2.2) it is easy to see that

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{G}\mathbf{G}^T + \Psi). \tag{2.4}$$

The key motivation for this model is that, because of the diagonality of Ψ, the observed variables $\mathbf{x}$ are conditionally independent given the latent variables, or factors, $\mathbf{y}$. The

intention is that the dependencies between the data variables $\mathbf{x}$ are explained by a smaller number of latent variables $\mathbf{y}$, while $\mathbf{n}$ represents variance unique to each observation variable. It can easily be shown [16] that the posterior in latent space is also normal,

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{y|x}, \mathbf{R}_{y|x}) \tag{2.5}$$

with mean

$$\begin{aligned}\boldsymbol{\mu}_{y|x} &= \mathbf{G}^T(\mathbf{G}\mathbf{G}^T + \Psi)^{-1}(\mathbf{x} - \boldsymbol{\mu}) \\ &= (\mathbf{I} + \mathbf{G}^T\Psi^{-1}\mathbf{G})^{-1}\mathbf{G}^T\Psi^{-1}(\mathbf{x} - \boldsymbol{\mu})\end{aligned}$$

and covariance

$$\mathbf{R}_{y|x} = (\mathbf{I} + \mathbf{G}^T\Psi^{-1}\mathbf{G})^{-1}. \tag{2.6}$$

The reduced dimensional representation of $\mathbf{x}$ is the posterior mean of $\mathbf{y}|\mathbf{x}$,

$$\widehat{\mathcal{M}}^{-1}(\mathbf{x}) = (\mathbf{I} + \mathbf{G}^T\Psi^{-1}\mathbf{G})^{-1}\mathbf{G}^T\Psi^{-1}(\mathbf{x} - \boldsymbol{\mu}) \tag{2.7}$$

which is the MMSE estimate of the latent variable $\mathbf{y}$ given $\mathbf{x}$. The dimensionality reduction mapping $\widehat{\mathcal{M}}^{-1}$ is linear and therefore smooth.

Given N i.i.d samples, the log-likelihood of the parameters $\boldsymbol{\Theta} = \{\mathbf{G}, \Psi, \boldsymbol{\mu}\}$ is the summation of the natural log of a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with covariance $\Sigma = \mathbf{G}\mathbf{G}^T + \Psi$,

$$l(\boldsymbol{\Theta}|\{\mathbf{x}_n\}_{n=1}^N) = -\frac{N}{2}(D\log 2\pi + \log|\Sigma| + \text{tr } \mathbf{S}\Sigma^{-1}) \tag{2.8}$$

where $\mathbf{S} = \frac{1}{N}\sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T$. The log-likelihood gradients are:

$$\nabla_{\boldsymbol{\mu}} l(\boldsymbol{\Theta}|\{\mathbf{x}_n\}_{n=1}^N) = -N\Sigma^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu}) \tag{2.9}$$

$$\nabla_{\mathbf{G}} l(\boldsymbol{\Theta}|\{\mathbf{x}_n\}_{n=1}^N) = -N(\Sigma^{-1}(\mathbf{I} - \mathbf{S}\Sigma^{-1})\mathbf{G}) \tag{2.10}$$

$$\nabla_{\Psi} l(\boldsymbol{\Theta}|\{\mathbf{x}_n\}_{n=1}^N) = -\frac{N}{2}\text{diag}(\Sigma^{-1}(\mathbf{I} - \mathbf{S}\Sigma^{-1})) \tag{2.11}$$

where $\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$. The maximum likelihood estimate of $\boldsymbol{\mu}$ is the sample mean, i.e. $\hat{\boldsymbol{\mu}}_{ML} = \bar{\mathbf{x}}$.

Remark 2.1:

If we apply an invertible linear transformation, $\mathbf{Q} \in \mathbb{R}^{d\times d}$, to the factors $\mathbf{y}$ to obtain a new set of factors $\tilde{\mathbf{y}} = \mathbf{Q}\mathbf{y}$, the prior distribution $f(\tilde{\mathbf{y}})$ is still normal, $\tilde{\mathbf{y}} \sim \mathcal{N}(0, \mathbf{Q}\mathbf{Q}^T)$, and the new mapping becomes $\mathbf{x} = \mathbf{G}\mathbf{Q}^{-1}\tilde{\mathbf{y}} + \boldsymbol{\mu}$ which leads to the same likelihood. That is, the new factor loadings become $\tilde{\mathbf{G}} = \mathbf{G}\mathbf{Q}^{-1}$. If $\mathbf{Q}$ is an arbitrary nonsingular matrix, which is called an oblique rotation of the factors [27], the new factors $\tilde{\mathbf{y}}$ will not be independent anymore. Moreover, the log-likelihood has infinitely many equivalent maxima resulting from orthogonal rotation of the factors in which the factors remain independent. Apart from these, it is not clear whether the log-likelihood has a unique global maximum or there exist sub-optimal solutions.

Remark 2.2:

Factor Analysis is covariant under component-wise rescaling of the data variables, i.e the scale factors simply become absorbed into rescaling of the noise variances, and the rows of $\mathbf{G}$ are rescaled by the same factors.

The parameters of a factor analysis model may be estimated using an EM algorithm.

1. **E-step**: Compute the posterior moments

$$\langle \mathbf{y} \rangle_n = \boldsymbol{\mu}_{y|x} \tag{2.12}$$

$$\langle \mathbf{y}\mathbf{y}^T \rangle_n = \mathbf{R}_{y|x} + \boldsymbol{\mu}_{y|x}\boldsymbol{\mu}_{y|x}^T \tag{2.13}$$

   for each data point $\mathbf{x}_n$ given the current parameter values $\mathbf{G}$ and Ψ. Note that $\langle\cdot\rangle$ represents posterior estimate $E[\cdot|\mathbf{x}]$.

2. **M-step**: Perform the following update equations for the factor loadings $\mathbf{G}$ and Ψ.

$$\mathbf{G} = \Big(\sum_{n=1}^{N}\mathbf{x}_n\langle \mathbf{y} \rangle_n\Big)\Big(\sum_{n=1}^{N}\langle \mathbf{y}\mathbf{y}^T \rangle_n\Big)^{-1} \tag{2.14}$$

$$\Psi = \frac{1}{N}\text{diag}(\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T - \mathbf{G}\langle \mathbf{y} \rangle_n \mathbf{x}_n^T) \tag{2.15}$$

where the updated moments are used and the "diag" operator sets all the off-diagonal elements of a matrix to zero.

Remark 2.3:

Factor Analysis estimate does not satisfy the condition that $\widehat{\mathcal{M}}^{-1} \circ \mathcal{M}$ be the identity, because $\mathbf{G}^T(\mathbf{G}\mathbf{G}^T + \Psi)^{-1}\mathbf{G} \neq \mathbf{I}$, except in the zero-noise case.

2.3 Probabilistic Principal Component Analysis

Certain links between factor analysis and PCA have been previously established, and such connections center on the special case of an isotropic error model, where $\Psi = \sigma^2\mathbf{I}$. In this case, maximum likelihood is equivalent to a least squares criterion and a principal component solution emerges in a straightforward manner. Assuming the correct choice of d, it is shown that both $\mathbf{G}$ and σ^2 can be estimated analytically through eigen-decomposition of $\mathbf{S}$, [28]. One limiting disadvantage of PCA is the absence of an associated probability density model. Deriving PCA from the perspective of density estimation would offer a number of important advantages. First, the corresponding likelihood would permit comparison with other density estimation techniques and facilitate statistical testing. Second, Bayesian inference methods could be applied (e.g., for model comparison) by combining the likelihood with a prior [18]. Third, the value of the probability density function could be used as a measure of the "degree of novelty" of a new data point.

Consider the factor analysis model of the previous section with an isotropic error model. i.e. $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathcal{M}(\mathbf{y}) = \mathbf{G}\mathbf{y} + \boldsymbol{\mu}$ and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. There exists a unique ML estimate closely related to the d principal components of the data. The log-likelihood function is given

as,

$$l(\boldsymbol{\Theta}|\{\mathbf{x}_n\}_{n=1}^N) = \log \prod_{n=1}^{N} f(\mathbf{x}_n) \tag{2.16}$$
$$= -\frac{N}{2}(D \log 2\pi + \log |\Sigma| + \text{tr } \mathbf{S}\Sigma^{-1})$$

where $f(\mathbf{x})$ is the normal pdf with mean, $\boldsymbol{\mu}$, and covariance, $\mathbf{G}^T\mathbf{G}+\sigma^2\mathbf{I}$ and $\Sigma = \sigma^2\mathbf{I}+\mathbf{G}\mathbf{G}^T$, and $\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T$. The likelihood function is maximized when $\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$ and the columns of $\mathbf{G}$ span the principal subspace of the data. The log-likelihood gradient w.r.t $\mathbf{G}$ is

$$\nabla_{\mathbf{G}} l(\boldsymbol{\Theta}|\{\mathbf{x}_n\}_{n=1}^N) = -N\Sigma^{-1}(\mathbf{I} - \mathbf{S}\Sigma^{-1})\mathbf{G} \tag{2.17}$$

where $\mathbf{S}$ is the sample covariance matrix. It can be shown [1] that the only non-zero stationary points occur for

$$\mathbf{G} = \mathbf{U}_d(\Lambda_d - \sigma^2\mathbf{I})^{1/2}\mathbf{Q} \tag{2.18}$$

where the d column vectors in the $D \times d$ matrix $\mathbf{U}_d$ are principal eigenvectors of $\mathbf{S}$, with corresponding eigenvalues in the $d \times d$ diagonal matrix Λ_d, and $\mathbf{Q}$ is an arbitrary $d \times d$ orthogonal rotation matrix. It may also be shown that the ML estimator for σ^2 is given by

$$\sigma^2 = \frac{1}{D-d}\sum_{i=d+1}^{D}\lambda_i \tag{2.19}$$

where λ_{d+1}, ..., λ_D are the smallest $D - d$ eigenvalues of $\mathbf{S}$ and so estimated σ^2 has a clear interpretation as the average variance "lost" per discarded dimensions. Therefore, the posterior in latent space is normal,

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{y|x}, \mathbf{R}_{y|x}) \tag{2.20}$$

with mean

$$\boldsymbol{\mu}_{y|x} = \mathbf{Q}^T\Lambda_d^{-1}(\Lambda_d - \sigma^2\mathbf{I})^{1/2}\mathbf{U}_d^T(\mathbf{x} - \boldsymbol{\mu}) \tag{2.21}$$

and covariance

$$\mathbf{R}_{y|x} = \sigma^2\mathbf{Q}^T\Lambda_d^{-1}\mathbf{Q}. \tag{2.22}$$

The MMSE estimate of the latent variable $\mathbf{y}$ given $\mathbf{x}$ is the posterior mean of $\mathbf{y}|\mathbf{x}$,

$$\widehat{\mathcal{M}}^{-1}(\mathbf{x}) = \mathbf{Q}^T \Lambda_d^{-1} (\Lambda_d - \sigma^2 \mathbf{I})^{1/2} \mathbf{U}_d^T (\mathbf{x} - \boldsymbol{\mu}). \tag{2.23}$$

Comparing this result to the extracted features obtained in PCA

$$\widehat{\mathcal{M}}^{-1}(\mathbf{x}) = \mathbf{U}_d^T (\mathbf{x} - \boldsymbol{\mu}), \tag{2.24}$$

we can conclude that the estimated features in PPCA are (1) element-wise scaled versions of those obtained in PCA, and (2) can be rotated without affecting the likelihood function.

Remark 2.4:

The most attractive property of PCA, is that it is the linear mapping that minimizes the least squares reconstruction error of a sample. Although, PPCA defines a probability model in the latent variable model, it is usually constructed as a reconstruction and dimensionality reduction technique. On the other hand, PPCA estimate does not satisfy the condition that $\widehat{\mathcal{M}}^{-1} \circ \mathcal{M}$ be the identity, because $\mathbf{G}^T (\mathbf{G}\mathbf{G}^T + \sigma^2 \mathbf{I})^{-1} \mathbf{G} \neq \mathbf{I}$, except in the zero-noise case. Therefore, to adapt reconstruction mapping of PPCA to PCA, Tipping and Bishop [2] change the reconstruction equation as follows,

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} (\sigma^2 \mathbf{I} + \mathbf{G}^T \mathbf{G}) \widehat{\mathcal{M}}^{-1}(\mathbf{x}) + \boldsymbol{\mu} \\ &= \mathbf{U}_d \mathbf{U}_d^T (\mathbf{x} - \boldsymbol{\mu}) + \boldsymbol{\mu}. \end{aligned}$$

which gives identical reconstruction error as with the conventional PCA. This is similar to the model bias in rank d reconstruction of stationary signals, discussed in [29])

Remark 2.5:

Since the noise model variance σ^2 is the same for all observed variables, the directions of the columns of $\mathbf{G}$ will be more influenced by the variables that have higher noise, unlike Factor Analysis, which should be able to separate the linear correlations from the noise.

Remark 2.6:

PCA and PPCA are covariant under rotations of the data variables, since the transformed noise covariance $\sigma^2\mathbf{Q}\mathbf{Q}^T$ will only be proportional to the unit matrix if $\mathbf{Q}$ is an orthogonal matrix. The same covariance property is shared by standard non probabilistic PCA since a rotation of the coordinates induces a corresponding rotation of the principal axes. Due to this ambiguity, this framework is suitable for subspace identification rather than factor identification which needs additional constraints.

The log-likelihood of the estimated parameters is

$$l(\mathbf{\Theta}|\{\mathbf{x}_n\}_{n=1}^N) = -\frac{N}{2}(D(1+\log 2\pi) + \log|\mathbf{S}| + (D-d)\log\frac{a}{g}) \tag{2.25}$$

where a and g are the arithmetic and geometric means of the $D-d$ smallest eigenvalues of the sample covariance matrix $\mathbf{S}$. Although there is an EM algorithm that finds the d principal components by maximizing the log-likelihood [2], the fastest way to perform PCA is via numerical singular value decomposition (SVD) [30]. Again, as in factor analysis, it is possible to orthogonally rotate the latent variables while keeping the same distribution.

Remark 2.7:

For a fixed data set, PCA has the property of additivity, to the extent that the principal components obtained using a latent space of dimension d are exactly the same as the ones obtained using a latent space of dimension $d-1$ plus a new, additional principal component. However, this additivity property does not necessarily hold for Factor Analysis. That is, the factors found by a factor analysis of order d are, in general, all different from those found by a factor analysis of order $d-1$. That means one can only talk about the joint collection of d factors or the linear subspace spanned by them.

2.4 Mixture of Probabilistic PCA

The other significant advantage of deriving PCA from the perspective of density estimation is that the single PCA model could be extended to a mixture of such models. Since PCA only defines a linear projection of the data, the scope of its application is necessarily somewhat

limited. This has naturally motivated various developments of nonlinear principal component analysis in an effort to retain a greater proportion of the variance using fewer components. An alternative paradigm are global nonlinear approaches, such as principal curves [24, 31], multi-layer auto-associative neural networks [32] and the generative topographic mapping, or GTM, [17] is to model nonlinear structure with a collection, or mixture, of local linear sub-models.

The association of a probability model with PCA, enable us to model complex data structures with a combination of local PCA models through the mechanism of a mixture of probabilistic principal component analyzers [1]. This mixture can be formalized as follows,

$$\mathbf{x} = \begin{cases} \mathbf{G}_1\mathbf{y} + \boldsymbol{\mu}_1 + \mathbf{n}_1 & \text{w.p. } \pi_1 \\ \mathbf{G}_2\mathbf{y} + \boldsymbol{\mu}_2 + \mathbf{n}_2 & \text{w.p. } \pi_2 \\ \qquad\qquad ... & \text{w.p. } \pi_l \\ \mathbf{G}_L\mathbf{y} + \boldsymbol{\mu}_L + \mathbf{n}_L & \text{w.p. } \pi_L \end{cases} \tag{2.26}$$

where π_l is the corresponding mixing proportion, with $\pi_l \geq 0$ and $\sum_{l=1}^{L} \pi_l = 1$. This formulation would permit all of the model parameters to be determined from ML, where both the appropriate partitioning of the data and the determination of the respective principal axes occur automatically as the likelihood is maximized. The log-likelihood of observing the data set for such a mixture model is

$$l(\boldsymbol{\Theta}|\{\mathbf{x}_n\}_{n=1}^{N}) = \sum_{n=1}^{N} \log f(\mathbf{x}_n), \tag{2.27}$$

where $f(\mathbf{x}_n) = \sum_{l=1}^{L} \pi_l f(\mathbf{x}_n|l)$ and $f(\mathbf{x}|l)$ is a single PPCA model, normal pdf with $\boldsymbol{\mu}_l$ mean and $\mathbf{G}_l\mathbf{G}_l^T + \sigma_l^2\mathbf{I}$ covariance. Note that a separate mean vector $\boldsymbol{\mu}_l$ is now associated with each of the L mixture components, along with the parameters $\mathbf{G}_l$ and σ_l^2. The corresponding generative model for the mixture case now requires the random choice of a mixture component according to the proportions π_l, followed by sampling from the $\mathbf{y}$ and $\mathbf{n}$ distributions and applying linear equation in the single model case, and using the appropriate parameters $\boldsymbol{\mu}_l$, $\mathbf{G}_l$, and σ_l^2.

Clustering and Data Reconstruction

Generating a local PCA model of the form illustrated above is often prompted by the ultimate goal of accurate data reconstruction. For the mixture of PPCA model, data points are assigned to the mixture components (in a soft fashion) according to the posterior distribution associated with mixture component, l,

$$R_{n,l} = f(l|\mathbf{x}_n) = \frac{f(\mathbf{x}_n|l)\pi_l}{f(\mathbf{x}_n)} \tag{2.28}$$

where $R_{n,l}$ is the probability of data sample $\mathbf{x}_n$ being generated by model l. It is also called *responsibility* of the mixture component l for the generating data sample $\mathbf{x}_n$. Since denominator $f(\mathbf{x}_n)$ is the same for all components, $R_{n,l} \propto \pi_l f(\mathbf{x}_n|l)$. In order to perform clustering for a data sample $\mathbf{x}_n$, MAP estimate of mixture component is used as follows,

$$\hat{l}_n = \underset{l}{\operatorname{argmax}}\, R_{n,l}. \tag{2.29}$$

Therefore, for a given data point $\mathbf{x}$, there is now an associated linear model, index of which is estimated as $\hat{l}$. The estimated latent variable in MPPCA model is given by

$$\widehat{\mathcal{M}}^{-1}(\mathbf{x}) = (\sigma_{\hat{l}}^2\mathbf{I} + \mathbf{G}_{\hat{l}}^T\mathbf{G}_{\hat{l}})^{-1}\mathbf{G}_{\hat{l}}^T(\mathbf{x} - \boldsymbol{\mu}_{\hat{l}}). \tag{2.30}$$

Similar to PPCA, in order to achieve a least squares (LS) estimate of data sample $\mathbf{x}$ the reconstruction rule must be written as,

$$\begin{aligned}\hat{\mathbf{x}} &= \mathbf{G}_{\hat{l}}(\mathbf{G}_{\hat{l}}^T\mathbf{G}_{\hat{l}})^{-1}(\sigma_{\hat{l}}^2\mathbf{I} + \mathbf{G}_{\hat{l}}^T\mathbf{G}_{\hat{l}})\widehat{\mathcal{M}}^{-1}(\mathbf{x}) + \boldsymbol{\mu}_{\hat{l}} \\ &= P_{G_{\hat{l}}}(\mathbf{x} - \boldsymbol{\mu}_{\hat{l}}) + \boldsymbol{\mu}_{\hat{l}}.\end{aligned}$$

where $P_{G_{\hat{l}}}$ is the projection matrix onto subspace $G_{\hat{l}}$.

Parameter Estimation

We can develop an iterative EM algorithm for optimization of all of the model parameters π_l, $\boldsymbol{\mu}_l$, $\mathbf{G}_l$, and σ_l^2. Using (2.28), it is shown [1] that we obtain the following parameter updates for π_l and $\boldsymbol{\mu}_l$:

$$\pi_l = \frac{1}{N}\sum_{n=1}^{N} R_{n,l} \tag{2.31}$$

$$\boldsymbol{\mu}_l = \frac{1}{\sum_{n=1}^{N} R_{n,l}} \sum_{n=1}^{N} R_{n,l}\mathbf{x}_n \tag{2.32}$$

Thus the updates for π_l and $\boldsymbol{\mu}_l$ correspond exactly to those of a standard Gaussian mixture formulation [22]. Furthermore, it is also shown [1] that the combination of the E- and M-steps leads to the intuitive result that the axes $\mathbf{G}_l$ and the noise variance σ_l^2 are determined from the local responsibility weighted covariance matrix:

$$\mathbf{S}_l = \frac{1}{\sum_{n=1}^{N} R_{n,l}} \sum_{n=1}^{N} R_{n,l}(\mathbf{x}_n - \boldsymbol{\mu}_l)(\mathbf{x}_n - \boldsymbol{\mu}_l)^T \tag{2.33}$$

by standard eigen-decomposition in exactly the same manner as for a single PPCA model. Iteration of equations (2.28), (2.31) and (2.32) in sequence followed by computation of $\mathbf{G}_l$ and σ_l^2, from (2.33) using (2.18) and (2.19) is guaranteed to find a local maximum of the log-likelihood in (2.27). At convergence, each weight matrix $\mathbf{G}_l$ spans the principal subspace of its respective $\mathbf{S}_l$.

Implementation of this method was shown [1] to yield good results in image compression and handwritten digit recognition applications, although the reconstruction error attained is in general larger than that attained by a non-probabilistic mixture of PCAs trained to minimize the reconstruction error. This is reasonable since the probabilistic mixture of PCAs is trained to maximize the log-likelihood rather than minimize the reconstruction error.

Remark 2.8:

Ghahramani and Hinton [33] construct a mixture of factor analyzers where each factor analyzer is characterized by two kinds of parameters, the mean vector $\boldsymbol{\mu}_l$ and the loadings matrix $\mathbf{G}_l$, in addition to the mixing proportion π_l. All analyzers share a common noise model diagonal covariance matrix Ψ for simplicity. They give an EM algorithm for estimating the parameters $\{\{\pi_l, \boldsymbol{\mu}_l, \mathbf{G}_l\}_{l=1}^{L}, \Psi\}$ by ML from a sample, but the parameters estimation in this case didn't show any resemblance with PCA.

Remark 2.9:

A mixture of diagonal Gaussians and a mixture of spherical Gaussians [1] can be seen, at limiting cases, as a mixture of factor analyzers with zero factors per component model and a mixture of principal component analyzers with zero principal components per component model, respectively. Thus, Gaussian mixtures explain the data by assuming that it is exclusively due to noise, without any underlying (linear) structure.

Remark 2.10:

This methods ignores class labels of data samples, i.e., it is unsupervised. In order to use MPPCA as a classification method, one needs to build a model of data samples for each class separately, and classify unseen data according to the model to which they are most "likely". In this sense, estimated features are not useful for discriminative purposes.

2.5 Conclusion

In this chapter, we reviewed three subspace latent variable models. We began with the most common example of a latent variable model, i.e. the statistical Factor Analysis. Different shortcomings of this method such as having multiple local likelihood maxima and lack of additivity property were discussed. With a simple change in Factor Analysis model, PPCA was then derived from the perspective of density estimation. The association of a probability model with PCA offers tempting prospects. First, being able to compare with other density estimation techniques and facilitate statistical testing through the resultant likelihood. Second, Bayesian inference methods could be applied by combining the likelihood with a prior. Third, the value of the probability density function could be used as a measure of the "degree of novelty" of a new data point. And finally, being able to model complex data structures with a combination of local PCA models through the mechanism of a mixture of probabilistic principal component analyzers. Modeling complexity in data by a combination of simple linear models is an attractive paradigm offering both computational and algorithmic advantages along with increased ease of interpretability.

The probabilistic model for PCA was exploited to combine local PCA models within the framework of a probabilistic mixture in which all the parameters are determined from ML using an EM algorithm. In addition to the clearly defined nature of the resulting algorithm, the primary advantage of this approach is the definition of an observation density model. A possible disadvantage of the probabilistic approach to combining local PCA models is that, by optimizing a likelihood function, the MPPCA does not directly minimize squared reconstruction error. For applications where this is the key requirement, algorithms which explicitly minimize reconstruction error should be expected to perform better. There is evidence that the smoothing implied by the soft clustering inherent in the MPPCA helps to reduce overfitting, particularly in the case of the experiments where the statistics of the test data set differed from the training data much more so than for other examples.

CHAPTER 3

MIXTURE OF FACTOR MODELS FOR JOINT DIMENSIONALITY REDUCTION AND CLASSIFICATION

3.1 Introduction

A powerful approach to probabilistic modeling involves supplementing the set of observed variables with additional latent variables. By defining a joint distribution over observable and latent variables, the corresponding distribution of the observed variables is then obtained by marginalization. This allows relatively complex distributions to be expressed in terms of more tractable joint distributions. One well-known example of a discrete hidden variable model is the mixture distribution in which the hidden variable is the component index. In the case of continuous latent variables, factor analysis is a famous example. This powerful approach inspired us to model labeled high-dimensional data using low-dimensional latent variables.

This chapter provides the general framework of the proposed model. The framework used here is inspired by the work in [1] which takes advantage of both discrete and continuous latent variables, by using a mixture of linear models leading to a nonlinear dimensionality reduction and clustering algorithm. In contrast, the proposed method provides a supervised probabilistic approach for analyzing labeled high dimensional data by exploiting a set of low dimensional latent variables. With the aid of these low-dimensional latent variables, a mixture of linear models is used to locally linearize the unknown nonlinear manifold that the data supposedly lie on. An optimum linear classifier is then built in the latent domain to separate the support of latent variables for each class and subsequently help to provide the discriminative features.

This chapter also develops optimal dimensionality reduction, clustering and reconstruction algorithms. Introducing hidden variables leads to a mixture of linear models in the

original data domain hence allowing us to derive the joint probability density of the data and latent variables. Bayes theorem is used to invert the transformation from latent space to data space, which leads to dimensionality reduction whereas estimating the discrete latent variable amounts to data clustering. Given estimates of the latent variables, this probabilistic approach provides a mechanism to traverse from latent to data space and perform data reconstruction.

Finally, we develop a Bayesian classification algorithm, based on data samples as well as the estimated low dimensional features. Introducing hidden variables allows us to derive the joint probability density of the data and labels variables as well.

This chapter is organized as follows: Section 3.2 provides a general framework of the proposed method. Section 3.3 is devoted to developing dimensionality reduction, data clustering, and reconstruction algorithms in our system. Classification is covered in Section 3.4. Concluding remarks are given in Section 3.5.

3.2 General Framework

In [2], Tipping and Bishop modified the regular Factor Analysis and proposed a probabilistic method for linear dimensionality reduction using latent variables. In their paper, an ML framework for estimating the parameters was shown to lead to the same results as PCA. In order to model complex data structures they proposed a combination of local linear PCA models leading to a mixture of probabilistic PCA (MPPCA), [1]. MPPCA offers an alternative dimensionality reduction algorithm with the added capability for clustering as well as data reconstruction. Here, we introduce a supervised statistical-based approach which accomplishes these tasks and also performs classification based upon either the data samples or the estimated latent variables.

Consider a binary classification problem where $\mathbf{x} \in \mathbb{R}^D$ represents the data observation vector and scalar variable $z \in \{0, 1\}$ represents its corresponding class label. Assume the data in the ambient space, regardless of its class label, can be represented by a set of linear

models. One way to model this complex structure is to introduce low-dimensional latent variable $\mathbf{y} \in \mathbb{R}^d$ $(d \ll D)$ which is assumed to be multi-variate normal with pdf $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The following set of linear models is then proposed to represent the original data in the ambient space,

$$\mathbf{x} = \begin{cases} \mathbf{G}_1\mathbf{y} + \boldsymbol{\mu}_1 + \mathbf{n}_1 & \text{w.p. } \pi_1 \\ \mathbf{G}_2\mathbf{y} + \boldsymbol{\mu}_2 + \mathbf{n}_2 & \text{w.p. } \pi_2 \\ \qquad\qquad ... & \text{w.p. } \pi_l \\ \mathbf{G}_L\mathbf{y} + \boldsymbol{\mu}_L + \mathbf{n}_L & \text{w.p. } \pi_L \end{cases} \tag{3.1}$$

where L is the number of the linear models and $\{\mathbf{n}_l\}_{l=1}^L$ are white Gaussian noise vectors with pdf $\mathcal{N}(\mathbf{0}, \sigma_l^2\mathbf{I})$. The assumed distribution for the noise components $\{\mathbf{n}_l\}_{l=1}^L$ implies that given latent variable, $\mathbf{y}$, and model index, $l \in \{1, ..., L\}$, the elements of the observation vector, $\mathbf{x}$, are mutually independent. Such a model may also be termed "generative", as data vectors

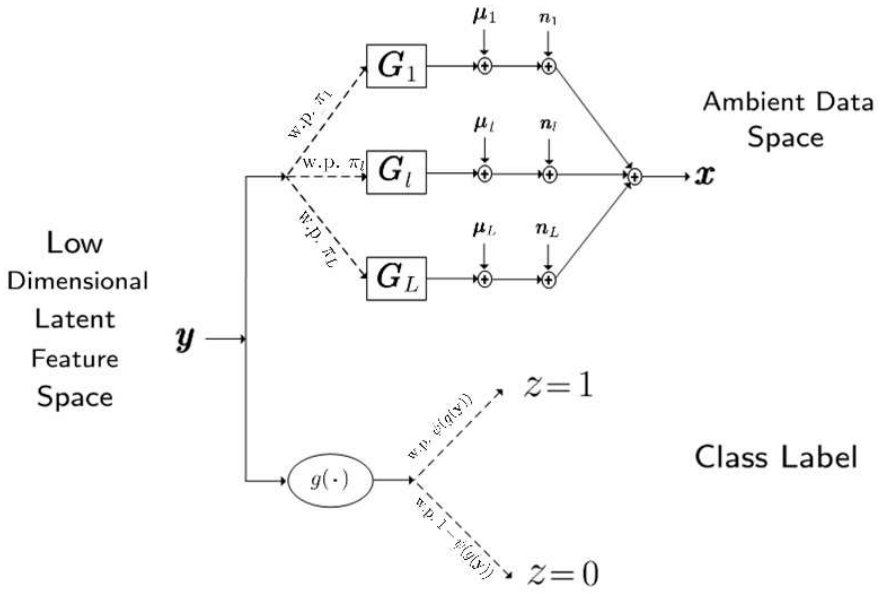

Figure 3.1: Model representation of the data and class label based upon latent variable vector $\mathbf{y}$.

$\mathbf{x}$ may be generated by sampling from the $\mathbf{y}$, l and $\mathbf{n}_l$ distributions and applying (3.1).

To build a joint dimensionality reduction and classification algorithm, we assume that the latent variable, $\mathbf{y}$, can be used as a representative feature vector to determine the class

label z of the original data $\mathbf{x}$. That is, we have

$$z = \begin{cases} 1 & \text{w.p. } \psi(g(\mathbf{y})) \\ 0 & \text{w.p. } 1 - \psi(g(\mathbf{y})) \end{cases} \tag{3.2}$$

where $g(\mathbf{y}) = \mathbf{w}^T\mathbf{y} + b$ is a linear discriminant function (DF) in the latent space and $\psi(\cdot)$ is an appropriate squashing function (e.g., CDF of standard normal distribution) which maps the output of DF to $[0, 1]$ range for decision-making. Figure 3.1 depicts this data representation model and class assignment process based upon latent variable $\mathbf{y}$.

It can easily be verified (see Section 3.3) that the data model in (3.1) and $\mathbf{y} \sim \mathcal{N}(\mathbf{0}; \mathbf{I})$ assumption result in a mixture of normal distributions with mixing weights π_l for the pdf of random vector $\mathbf{x}$. Additionally, it is interesting to note that although the distribution of latent variable, irrespective of class, is assumed to be normal $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the distribution of $\mathbf{y}$ in each class $z = 0, 1$ is, what we refer to a squashed multi-variate normal distribution. This is shown next.

Proposition 3.1:

a. The distribution of the latent variable, $\mathbf{y}$, given binary random variable z (class label) is a squashed multivariate normal,

$$f(\mathbf{y}|z) = f(\mathbf{y}) \Big(\frac{\psi(g(\mathbf{y}))}{E[\psi(g(\mathbf{y}))]} \Big)^z \Big(\frac{1 - \psi(g(\mathbf{y}))}{1 - E[\psi(g(\mathbf{y}))]} \Big)^{1-z}, \tag{3.3}$$

where $\psi(.)$ is an arbitrary squashing function with $\psi(g(\mathbf{y})) = f(z = 1|\mathbf{y})$ and $g(\mathbf{y}) = 0$ specifies the boundaries. For the special case of Heaviside function, i.e. $\psi(.) = \mathbf{1}_{(0,+\infty)}(.)$, this model coincide with the general Truncated Multivarite Normal Distribution (see Appendix A, section A.1).

b. If $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{R})$, in special case of $g(\mathbf{y}) = \mathbf{w}^T\mathbf{y} + b$ and $\psi(.)$ being the CDF of standard normal distribution, $\Phi(.)$, the first and second order statistics of this distribution are given by

$$E[\mathbf{y}|z] = \boldsymbol{\mu} + \gamma(z)\mathbf{R}\mathbf{w} \tag{3.4}$$

$$E[\mathbf{y}\mathbf{y}^T|z] = \mathbf{R} + \boldsymbol{\mu}\boldsymbol{\mu}^T + \gamma(z)(\mathbf{R}\mathbf{w}\boldsymbol{\mu}^T + \boldsymbol{\mu}\mathbf{w}^T\mathbf{R} - \frac{\mathbf{w}^T\boldsymbol{\mu} + b}{1 + \mathbf{w}^T\mathbf{R}\mathbf{w}}\mathbf{R}\mathbf{w}\mathbf{w}^T\mathbf{R}) \tag{3.5}$$

where

$$\gamma(z) = \frac{\phi\Big(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}\mathbf{w}}}\Big)}{\Big(\Phi\Big(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}\mathbf{w}}}\Big) - \mathbf{1}_{\{0\}}(z)\Big)\sqrt{1 + \mathbf{w}^T\mathbf{R}\mathbf{w}}},$$

where $\phi(\cdot)$ is the standard normal pdf and $\mathbf{1}_A(z)$ is the indicator function,

$$\mathbf{1}_A(z) = \begin{cases} 1 & \text{for} \quad z \in A \\ 0 & \text{for} \quad z \notin A. \end{cases}$$

Proof of Proposition 3.1: See Appendix A, section A.2.

The class conditional distributions $f(\mathbf{y}|z = 1)$ and $f(\mathbf{y}|z = 0)$ have almost disjoint supports (see Figure 3.2), i.e. this model almost separates the support of $\mathbf{y}$ for each class.

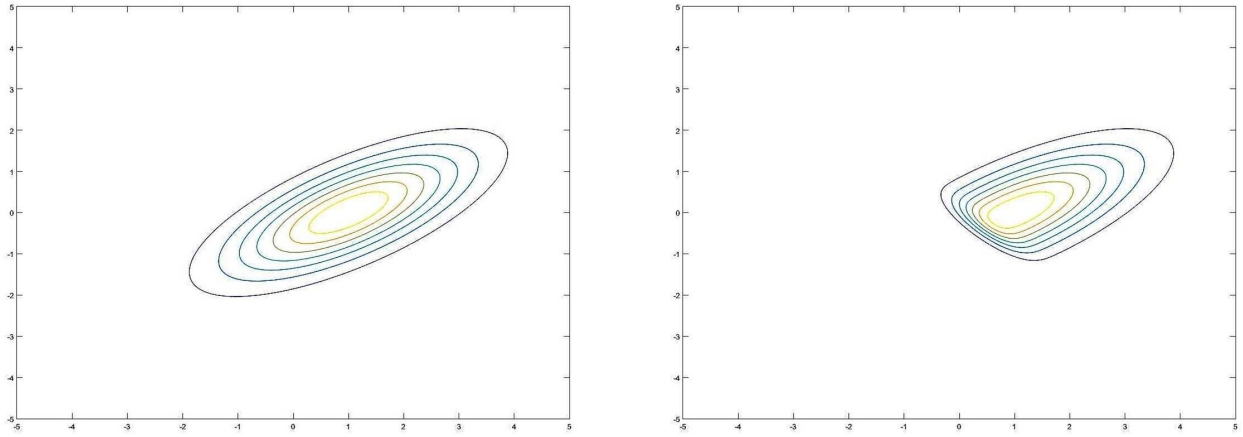

Figure 3.2: Left: Contours of a Multivariate Normal Distribution. Right: Contours of a Linearly Squashed Multivariate Normal Distribution for $z = 1$ or 0.

Given training dataset $\mathcal{D} = \{\mathbf{x}_n, z_n\}_{n=1}^N$, we would like to estimate matrices $\mathbf{G}_l$, mean vectors $\boldsymbol{\mu}_l$, variances σ_l^2, priors π_l, weight vector $\mathbf{w}$ and bias b using the ML estimation. In Chapter 4, we shall show how a subset of parameters can be estimated using the EM algorithm whereas the rest can be estimated using a steepest descent method. However, in the following section we assume these parameters are already computed during the training

phase and show how we can perform dimensionality reduction and classification using our framework.

3.3 Dimensionality Reduction, Clustering and Reconstruction from a known Mixture Model

For dimensionality reduction using this algorithm, one needs to estimate a low dimensional latent variable $\mathbf{y} \in \mathbb{R}^d$ for an observation $\mathbf{x} \in \mathbb{R}^D$. The solution of this problem is obtained by minimizing the conditional risk of dimensionality reduction,

$$\mathcal{R}_{dim}(\mathbf{x}) = E[\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})|\mathbf{x}],$$

where $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ is a quadratic loss function between the latent variable, $\mathbf{y}$, and its associated estimate $\hat{\mathbf{y}}$. This minimization leads to the Minimum Mean Squared Error (MMSE) estimate of $\mathbf{y}$ or $\hat{\mathbf{y}} = E[\mathbf{y}|\mathbf{x}]$. The following theorem gives the solution to this problem.

Theorem 3.1: Analysis Equation

Given the observation $\mathbf{x}_n \in \mathbb{R}^D$ in the ambient space, the MMSE estimate of $\mathbf{y}_n \in \mathbb{R}^d$ is given by,

$$\hat{\mathbf{y}}_n = \sum_{l=1}^{L} f(l|\mathbf{x}_n)\boldsymbol{\mu}_{y|x_n,l} \tag{3.6}$$

where $\boldsymbol{\mu}_{y|x_n,l} = (\sigma_l^2\mathbf{I}+\mathbf{G}_l^T\mathbf{G}_l)^{-1}\mathbf{G}_l^T(\mathbf{x}_n-\boldsymbol{\mu}_l)$ and $f(l|\mathbf{x}_n) = \frac{\pi_l f(\mathbf{x}_n|l)}{f(\mathbf{x}_n)}$ is the posterior probability of model index. Note that $f(\mathbf{x}|l)$ is the pdf of a normal distribution with mean $\boldsymbol{\mu}_{x|l} = \boldsymbol{\mu}_l$ and covariance $\mathbf{R}_{x|l} = \sigma_l^2\mathbf{I} + \mathbf{G}_l\mathbf{G}_l^T$, and $f(\mathbf{x})$ can be computed using $f(\mathbf{x}) = \sum_{l=1}^{L} \pi_l f(\mathbf{x}|l)$. For this estimate, the minimum risk of dimensionality reduction is

$$R^*_{dim.}(\mathbf{x}_n) = \sum_{l=1}^{L} f(l|\mathbf{x}_n)(\mathrm{tr}\{\mathbf{R}_{y|x,l}\} + \|\boldsymbol{\mu}_{y|x_n,l} - \hat{\mathbf{y}}_n\|_2^2). \tag{3.7}$$

Proof of Theorem 3.1:

To evaluate the MMSE estimate of the latent variable, $\hat{\mathbf{y}} = E[\mathbf{y}|\mathbf{x}]$ requires finding the *a posteriori* distribution $f(\mathbf{y}|\mathbf{x})$, which can be related to $f(\mathbf{y}, l|\mathbf{x})$ by,

$$f(\mathbf{y}|\mathbf{x}) = \sum_{l=1}^{L} f(\mathbf{y}, l|\mathbf{x}). \tag{3.8}$$

However, using the chain rule $f(\mathbf{y}, l|\mathbf{x}) = f(l|\mathbf{x})f(\mathbf{y}|\mathbf{x}, l)$, where $f(l|\mathbf{x}) = \frac{\pi_l f(\mathbf{x}|l)}{f(\mathbf{x})}$ is the posterior distribution of model index, l, and $f(\mathbf{y}|\mathbf{x}, l)$ is expressed as

$$f(\mathbf{y}|\mathbf{x}, l) = \frac{f(\mathbf{x}, \mathbf{y}|l)}{f(\mathbf{x}|l)}. \tag{3.9}$$

That is, finding $f(\mathbf{y}|\mathbf{x})$ requires the knowledge of $f(\mathbf{x}|l)$, $f(\mathbf{x})$, and $f(\mathbf{x}, \mathbf{y}|l)$. Now, given the model index, l, using the locally linear model in (3.1) (see Figure 3.1) we have,

$$\mathbf{x} = \mathbf{G}_l\mathbf{y} + \boldsymbol{\mu}_l + \mathbf{n}_l, \tag{3.10}$$

where $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{n}_l \sim \mathcal{N}(\mathbf{0}, \sigma_l^2\mathbf{I})$ which makes $f(\mathbf{x}|l)$ a normal distribution with mean $\boldsymbol{\mu}_{x|l} = \boldsymbol{\mu}_l$ and covariance $\mathbf{R}_{x|l} = \sigma_l^2\mathbf{I} + \mathbf{G}_l\mathbf{G}_l^T$. Furthermore, using $f(\mathbf{x}) = \sum_{l=1}^{L} \pi_l f(\mathbf{x}|l)$ implies that $\mathbf{x}$ has pdf which is a mixture of normal distributions with weights π_l. Finally, $f(\mathbf{x}, \mathbf{y}|l)$ is also normal with the following composite mean

$$\boldsymbol{\mu}_{x,y|l} = \begin{bmatrix} \boldsymbol{\mu}_l \\ \mathbf{0} \end{bmatrix}, \tag{3.11}$$

and composite covariance matrix

$$\mathbf{R}_{x,y|l} = \begin{bmatrix} \mathbf{G}_l\mathbf{G}_l^T + \sigma_l^2\mathbf{I} & \mathbf{G}_l \\ \mathbf{G}_l^T & \mathbf{I} \end{bmatrix}. \tag{3.12}$$

Using these results in (3.9), it can easily be shown that $f(\mathbf{y}|\mathbf{x}, l)$ is normal with mean

$$\begin{aligned} \boldsymbol{\mu}_{y|x,l} &= \mathbf{G}_l^T(\sigma_l^2\mathbf{I} + \mathbf{G}_l\mathbf{G}_l^T)^{-1}(\mathbf{x} - \boldsymbol{\mu}_l) \\ &= (\sigma_l^2\mathbf{I} + \mathbf{G}_l^T\mathbf{G}_l)^{-1}\mathbf{G}_l^T(\mathbf{x} - \boldsymbol{\mu}_l), \end{aligned} \tag{3.13}$$

and covariance

$$\begin{aligned} \mathbf{R}_{y|x,l} &= \mathbf{I} - \mathbf{G}_l^T(\sigma_l^2\mathbf{I} + \mathbf{G}_l\mathbf{G}_l^T)^{-1}\mathbf{G}_l \\ &= (\mathbf{I} + \frac{1}{\sigma_l^2}\mathbf{G}_l^T\mathbf{G}_l)^{-1}. \end{aligned} \tag{3.14}$$

Thus, given the observation $\mathbf{x}_n$, the MMSE estimate of $\mathbf{y}_n$ may be written as

$$\hat{\mathbf{y}}_n = \sum_{l=1}^{L} f(l|\mathbf{x}_n)\boldsymbol{\mu}_{y|x_n,l},$$

where $\boldsymbol{\mu}_{y|x_n,l}$ is obtained using (3.13). Since (3.6) provides an unbiased estimate of $\mathbf{y}$, the minimum risk of dimensionality reduction is $R^*_{dim.}(\mathbf{x}) = \text{tr}\{\text{Cov}[\mathbf{y}|\mathbf{x}]\}$ which can be written as

$$\begin{aligned} R^*_{dim.}(\mathbf{x}_n) &= \text{tr}\{\text{Cov}[\mathbf{y}|\mathbf{x}_n]\} \\ &= \text{tr}\{E[\mathbf{y}\mathbf{y}^T|\mathbf{x}_n]\} - \hat{\mathbf{y}}_n^T\hat{\mathbf{y}}_n \\ &= \sum_{l=1}^{L} f(l|\mathbf{x}_n)(\text{tr}\{E[\mathbf{y}\mathbf{y}^T|\mathbf{x}_n, l]\} - \hat{\mathbf{y}}_n^T\hat{\mathbf{y}}_n) \\ &= \sum_{l=1}^{L} f(l|\mathbf{x}_n)(\text{tr}\{\mathbf{R}_{y|x,l}\} + \boldsymbol{\mu}_{y|x_n,l}^T\boldsymbol{\mu}_{y|x_n,l} - \hat{\mathbf{y}}_n^T\hat{\mathbf{y}}_n) \end{aligned}$$

which yields (3.7) by using (3.6).

Remark 3.1:

It is interesting to note that since for a given observation $\mathbf{x}_n$ the model index is unknown, $\hat{\mathbf{y}}_n$ is a linear combination of the weighted LMMSE estimates of $\mathbf{y}_n$'s for a given model index l with $f(l|\mathbf{x}_n) = \frac{\pi_l f(\mathbf{x}_n|l)}{f(\mathbf{x}_n)}$ as weights. In other words, $f(l|\mathbf{x}_n)$ is indicative of model index likelihood and $\boldsymbol{\mu}_{y|x_n,l}$ is estimate of $\mathbf{y}_n$ given that model index. This implementation is depicted in Figure 3.3.

For data clustering and reconstruction using this algorithm,, we need to estimate the model index, l, for an observation $\mathbf{x}$. The associated conditional Bayes risk is

$$\mathcal{R}_{clus}(\mathbf{x}) = E[\mathcal{L}(l, \hat{l})|\mathbf{x}],$$

where l is the model (or cluster) index associated with $\mathbf{x}$, and $\hat{l}$ is an estimate of l. For the $0-1$ loss function $\mathcal{L}(l, \hat{l}) = 1 - \mathbf{1}_{\{\hat{l}\}}(l)$, the optimal clustering algorithm leads to the Maximum A Posteriori (MAP) estimate of l.

Theorem 3.2: Synthesis Equation

Given observation vector $\mathbf{x}_n$, for data clustering the MAP estimate of l gives,

$$\hat{l}_n = \underset{l}{\text{argmax}} \ \ \pi_l f(\mathbf{x}_n|l) \tag{3.15}$$

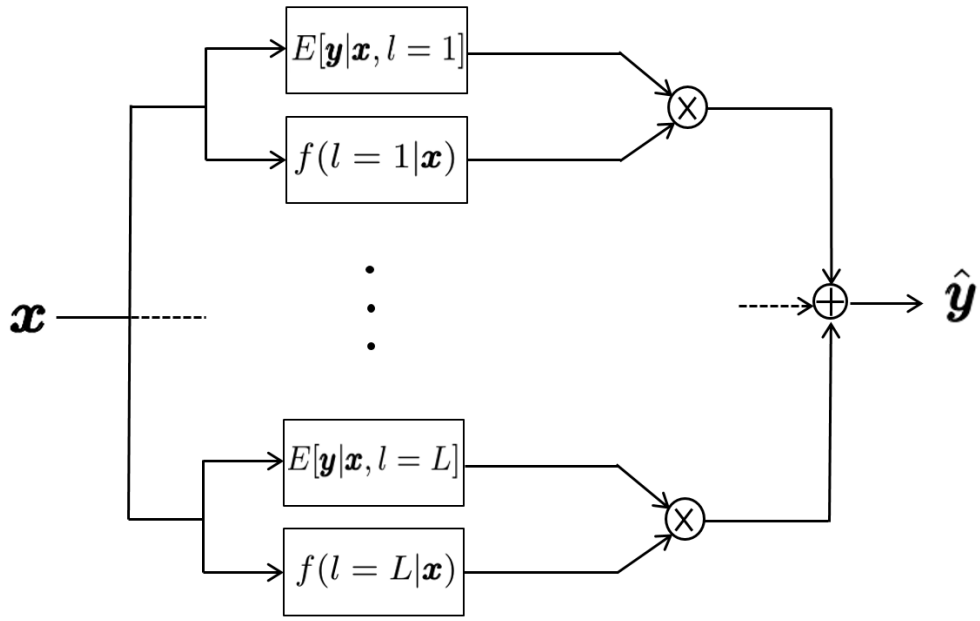

Figure 3.3: Dimensionality reduction process.

which can be used together with the estimate of the latent vector $\hat{\mathbf{y}}_n$ in the following equation to reconstruct the data from its low dimensional representation,

$$\hat{\mathbf{x}}_n = \mathbf{G}_{\hat{l}_n}\hat{\mathbf{y}}_n + \boldsymbol{\mu}_{\hat{l}_n}. \tag{3.16}$$

Proof of Theorem 3.2:

For the given $0-1$ risk function, the model index of each sample point can optimally be estimated using the *a posteriori* distribution of l given the observation $\mathbf{x}_n$,

$$\hat{l}_n = \underset{l}{\text{argmax}} \;\; f(l|\mathbf{x} = \mathbf{x}_n).$$

where $f(l|\mathbf{x} = \mathbf{x}_n) = \frac{\pi_l f(\mathbf{x}_n|l)}{f(\mathbf{x}_n)}$. However, since the denominator is independent of l, this MAP estimate gives (3.15). This estimate leads to the minimum clustering risk of

$$R^*_{clus.}(\mathbf{x}_n) = 1 - f(\hat{l}_n|\mathbf{x}_n). \tag{3.17}$$

Once the estimate of the model index (or cluster) is generated, the original data can be

reconstructed from the estimate of the latent variable $\hat{\mathbf{y}}_n$ using ad-hoc mapping,

$$\begin{aligned}\hat{\mathbf{x}}_n &= E[\mathbf{x}|\mathbf{y}, l]\big|_{\mathbf{y}=\hat{\mathbf{y}}_n, l=\hat{l}_n} \\ &= \mathbf{G}_{\hat{l}_n}\hat{\mathbf{y}}_n + \boldsymbol{\mu}_{\hat{l}_n}.\end{aligned}$$

Remark 3.2:

An important feature of the proposed method is that it allows for both analysis and synthesis of out-of-sample data as well as building an optimum classifier in the latent variable domain (see the next section).

Remark 3.3:

For very small clustering risk (i.e. it is highly probable that the data sample $\mathbf{x}_n$ belongs to a specific linear model), dimensionality reduction may simply be accomplished using

$$\hat{\mathbf{y}}_n = (\sigma^2_{\hat{l}_n}\mathbf{I} + \mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n})^{-1}\mathbf{G}^T_{\hat{l}_n}(\mathbf{x}_n - \boldsymbol{\mu}_{\hat{l}_n}). \tag{3.18}$$

Small clustering risk implies that the data sample $\mathbf{x}_n$ belongs to a specific linear model, e.g. for $L = 1$ case, clustering risk is zero. In this case, in order to have the least squares estimate of $\mathbf{x}_n$ without modifying the estimate of latent variable, $\hat{\mathbf{y}}_n$, reconstruction must be modified to

$$\begin{aligned}\hat{\mathbf{x}}_n &= \mathbf{G}_{\hat{l}_n}(\mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n})^{-1}(\sigma^2_{\hat{l}_n}\mathbf{I} + \mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n})\hat{\mathbf{y}}_n + \boldsymbol{\mu}_{\hat{l}_n} \\ &= P_{\mathbf{G}_{\hat{l}_n}}(\mathbf{x}_n - \boldsymbol{\mu}_{\hat{l}_n}) + \boldsymbol{\mu}_{\hat{l}_n}\end{aligned}$$

where $P_{\mathbf{G}_{\hat{l}_n}} = \mathbf{G}_{\hat{l}_n}(\mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n})^{-1}\mathbf{G}^T_{\hat{l}_n}$ is the projection matrix onto subspace $\mathbf{G}_{\hat{l}_n}$.

3.4 Classification

Given an observation $\mathbf{x} \in \mathbb{R}^D$ generated according to the model in Figure 3.1, the associated conditional risk for classification is

$$\mathcal{R}_{class}(\mathbf{x}) = E[\mathcal{L}(z, \hat{z})|\mathbf{x}],$$

where $z \in \{0,1\}$ is the class label associated with the observation $\mathbf{x}$, and $\hat{z} \in \{0,1\}$ is an estimate of z. For the $0-1$ loss function $\mathcal{L}(z,\hat{z}) = 1 - \mathbf{1}_{\hat{z}}(z)$, the optimal classifier that minimizes this risk (also known as Bayes Classifier) gives the MAP estimate of z.

Theorem 3.3:

Given an observation vector $\mathbf{x}_n \in \mathbb{R}^D$, the MAP estimate of the class label z_n is

$$\hat{z}_n = \mathbf{1}_{(\frac{1}{2},+\infty)}\Big(\sum_{l=1}^{L} f(l|\mathbf{x}_n) E[\psi(g(\mathbf{y}))|\mathbf{x}_n, l]\Big), \tag{3.19}$$

where $E[\psi(g(\mathbf{y}))|\mathbf{x}_n, l]$ is the MMSE estimate of z_n given the model index, l and $f(l|\mathbf{x}_n) = \frac{\pi_l f(\mathbf{x}_n|l)}{f(\mathbf{x}_n)}$ is the posterior distribution of model index, l, computed at $\mathbf{x}_n$.

Proof of Theorem 3.3:

For the given $0-1$ risk function, the class label of unknown sample point, $\mathbf{x}_n$, can be estimated using the MAP estimate of z given the observation $\mathbf{x}_n$,

$$\hat{z}_n = \underset{z\in\{0,1\}}{\operatorname{argmax}} \; f(z|\mathbf{x} = \mathbf{x}_n),$$

which requires the *a posteriori* distribution $f(z|\mathbf{x})$ or equivalently the joint distribution $f(\mathbf{x}, z)$. The latter can be obtained by marginalizing $f(\mathbf{x}, \mathbf{y}, z)$ over $\mathbf{y}$. On the other hand we have

$$f(\mathbf{x}, z) = \int f(\mathbf{x}, \mathbf{y}, z) d\mathbf{y}. \tag{3.20}$$

we can derive the joint distribution of $(\mathbf{x}, \mathbf{y}, z)$,

$$\begin{aligned} f(\mathbf{x}, \mathbf{y}, z) &= f(\mathbf{x}, \mathbf{y}) f(z|\mathbf{x}, \mathbf{y}) \\ &= f(\mathbf{x}, \mathbf{y}) f(z|\mathbf{y}) \\ &= f(\mathbf{x}, \mathbf{y}) \Big(\psi(g(\mathbf{y}))\Big)^z \Big(1 - \psi(g(\mathbf{y}))\Big)^{1-z}. \end{aligned} \tag{3.21}$$

Note that here we used the fact that given $\mathbf{y}$, the random variable z is independent of $\mathbf{x}$ (see Section 3.2 and Figure 3.1) and further $f(z|\mathbf{y}) = \psi(g(\mathbf{y}))^z \Big(1 - \psi(g(\mathbf{y}))\Big)^{1-z}$. The joint distribution of $\mathbf{x}$ and $\mathbf{y}$ in (3.21) can also be expanded as

$$f(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^{L} \pi_l f(\mathbf{x}|l) f(\mathbf{y}|\mathbf{x}, l). \tag{3.22}$$

Therefore, using $f(\mathbf{x}, z) = \int f(\mathbf{x}, \mathbf{y}, z) d\mathbf{y}$, (3.22), and (3.21) we have

$$f(\mathbf{x}, z) = \sum_{l=1}^{L} \pi_l f(\mathbf{x}, z|l) \tag{3.23}$$

$$= \sum_{l} \pi_l f(\mathbf{x}|l) \Big(E[\psi(g(\mathbf{y}))|\mathbf{x}, l] \Big)^z \Big(1 - E[\psi(g(\mathbf{y}))|\mathbf{x}, l] \Big)^{1-z}. \tag{3.24}$$

Finally, using $f(l|\mathbf{x}) = \frac{\pi_l f(\mathbf{x}|l)}{f(\mathbf{x})}$ we get

$$f(z|\mathbf{x}) = \sum_{l=1}^{L} f(l|\mathbf{x}) \Big(E[\psi(g(\mathbf{y}))|\mathbf{x}, l] \Big)^z \Big(1 - E[\psi(g(\mathbf{y}))|\mathbf{x}, l] \Big)^{1-z},$$

Thus, given the observation $\mathbf{x}_n$, the MAP estimate of the class label is

$$\hat{z}_n = \mathbf{1}_{(\frac{1}{2}, +\infty)} \Big(\sum_{l=1}^{L} f(l|\mathbf{x}_n) E[\psi(g(\mathbf{y}))|\mathbf{x}_n, l] \Big),$$

where $E[\psi(g(\mathbf{y}))|\mathbf{x}, l]$ is the MMSE estimate of z given the model index, l. Because the model index is unknown, MMSE estimate of z is a linear combination of the weighted MMSE estimates with $f(l|\mathbf{x})$ as weights. In other words, $f(l|\mathbf{x})$ is indicative of model index likelihood and $E[\psi(g(\mathbf{y}))|\mathbf{x}, l]$ is the estimate of z given that model index. Figure 3.4 illustrates the implementation of the classification process using our method. This estimate

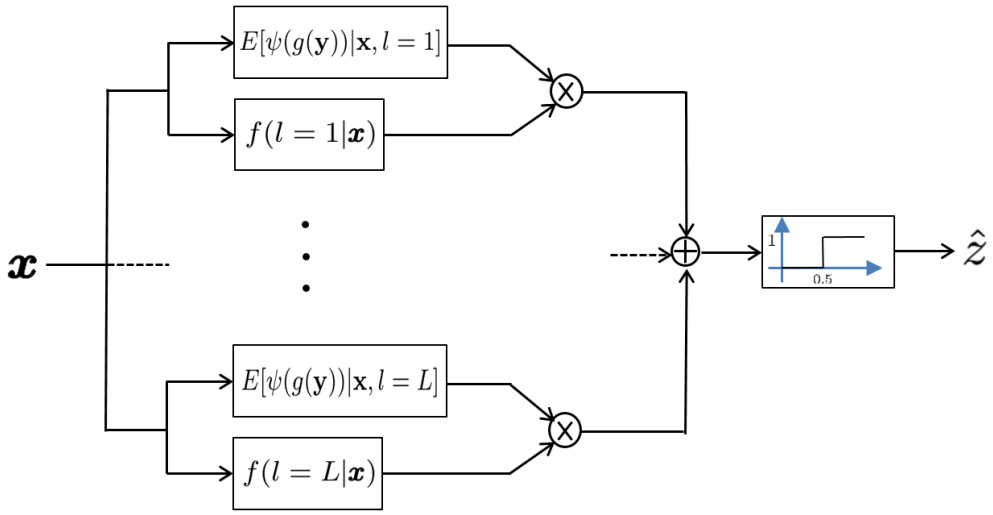

Figure 3.4: Classification process.

leads to minimum classification risk of

$$R^*_{clas.}(\mathbf{x}_n) = 1 - f(\hat{z}_n|\mathbf{x}_n). \tag{3.25}$$

Proposition 3.2:

For the linear DF, $g(\mathbf{y}) = \mathbf{w}^T\mathbf{y} + b$, and squashing function being the CDF of normal distribution, $\Phi(\cdot)$, (3.24) and (3.19) become

$$f(\mathbf{x}, z) = \sum_{l=1}^{L} \pi_l f(\mathbf{x}|l)\Phi\Big(\frac{\mathbf{w}^T\boldsymbol{\mu}_{y|x_n,l} + b}{\sqrt{1 + \mathbf{w}^T\mathbf{R}_{y|x,l}\mathbf{w}}}\Big)^z \Big(1 - \Phi\Big(\frac{\mathbf{w}^T\boldsymbol{\mu}_{y|x_n,l} + b}{\sqrt{1 + \mathbf{w}^T\mathbf{R}_{y|x,l}\mathbf{w}}}\Big)\Big)^{1-z}, \tag{3.26}$$

and

$$\hat{z}_n = \mathbf{1}_{(\frac{1}{2},+\infty)}\Big(\sum_{l=1}^{L} f(l|\mathbf{x}_n)\Phi\Big(\frac{\mathbf{w}^T\boldsymbol{\mu}_{y|x_n,l} + b}{\sqrt{1 + \mathbf{w}^T\mathbf{R}_{y|x,l}\mathbf{w}}}\Big)\Big). \tag{3.27}$$

Proof of Proposition 3.2: The closed form solution for $E[\Phi(\mathbf{w}^t\mathbf{y}+b)]$ expression is found in Appendix A. In order to find $E[\Phi(\mathbf{w}^t\mathbf{y} + b)|\mathbf{x}, l]$, one need to place the posterior mean, $\boldsymbol{\mu}_{y|x,1}$, and covariance, $\mathbf{R}_{y|x,l}$, in the final expressions in (3.24) and (3.19).

Remark 3.4:

For very small clustering risk, classification may simply be accomplished using

$$\hat{z}_n = \mathbf{1}_{(\frac{1}{2},+\infty)}(E[\psi(g(\mathbf{y}))|\mathbf{x}_n, \hat{l}_n]).$$

Using linear DF, $g(\mathbf{y}) = \mathbf{w}^t\mathbf{y}+b$, and normal CDF, $\Phi(.)$, as squashing function, classification decision is made using,

$$\hat{z}_n = \mathbf{1}_{(0,+\infty)}(\mathbf{w}^T\hat{\mathbf{y}}_n + b). \tag{3.28}$$

where $\mathbf{y}_n$ is generated using (3.18).

Remark 3.5:

Since the latent variable $\mathbf{y}$ carries everything to be known about the class of $\mathbf{x}$, the class label can be obtained by maximizing the *a posteriori* distribution of z given the latent variable $\mathbf{y}$, i.e

$$\hat{z} = \underset{z}{\text{argmax}}\ \ f(z|\mathbf{y}).$$

where $f(z|\mathbf{y})$ was defined before. Thus, the estimate of the class label for latent variable $\mathbf{y}$ is obtained using,

$$\hat{z} = \mathbf{1}_{(\frac{1}{2},+\infty)}(\psi(g(\mathbf{y})). \tag{3.29}$$

Now, if the risk of dimensionality reduction for an observed sample, $\mathbf{x}_n$, is small enough or alternatively, $\hat{\mathbf{y}}_n$, is accurate enough, for the special case of linear DF and using normal CDF as the squashing function, (3.29) yields the same result as in Remark 3.4 where $\hat{\mathbf{y}}_n$ is used instead of the actual latent variable vector $\mathbf{y}$ to perform classification for a given observation $\mathbf{x}_n$. For this case, the minimum classification risk can be given as

$$R^*_{clas.}(\hat{\mathbf{y}}_n) = 1 - f(\hat{z}_n|\hat{\mathbf{y}}_n). \tag{3.30}$$

Remark 3.6: Ambiguity

Both the joint distribution $f(\mathbf{x}, z)$ and estimate $\hat{z}_n$ in *Proposition 3.2* are functions of $\mathbf{G}_l\mathbf{w}$ and $\mathbf{G}_l\mathbf{G}_l^T$ as evident from equations for $\boldsymbol{\mu}_{y|x_n,l}$ and $\mathbf{R}_{y|x,l}$ in (3.13) and (3.14), respectively. This implies there is an inherent ambiguity in parameter estimation, since $\mathbf{G}_l\mathbf{U}$ and $\mathbf{U}^T\mathbf{w}$ for an arbitrary unitary matrix $\mathbf{U} \in \mathbb{R}^{d\times d}$ lead to the same results. However, the role of this unitary matrix is equivalent to rotating all the latent variable coordinates for all the samples which wont have any impact on the classification results. Nevertheless, we can simplify the model by fixing the direction of $\mathbf{w}$ and yet achieve the same likelihood and classification rule, i.e. $\mathbf{w} = \|\mathbf{w}\|\mathbf{e}_{fix}$. We can reduce the scope of redundant parameters by fixing the direction of weight vector $\mathbf{w}$ and estimating $\|\mathbf{w}\|$ instead.

3.5 Conclusion

In this chapter, the general framework of the proposed method was discussed. A mixture of factor models relates latent variables to observable variables, i.e. a mixture of linear models is defined to map latent variables to ambient data space. On the other hand, the pdf of the low dimensional latent variable associated with each class was a squashed multivariate normal leading to linear separability of the estimated low-dimensional latent variables (features).

Estimating the continuous latent variable $\mathbf{y}$ for a given data sample amounts to dimensionality reduction. It was shown that resultant low dimensional feature is a collaborative estimates of each linear model weighted by likelihood of the model. On the other hand, estimating model index l amounts to data clustering and subsequent data reconstruction.

In the special case of small clustering risk or when $L = 1$, a simplified least squares (LS) version of reconstruction is provided to reconstruct data samples.

It is shown that classification in our system can be accomplished by estimating class label z for either a given unknown data sample or its estimated latent variable. Estimated class label based on actual data sample is a collaborative estimates of each linear model weighted by likelihood of the model. Due to its optimality, which leads to the minimum classification risk, it is preferable to estimate class labels based on data samples. However, small clustering or small dimensionality reduction risk, allows us to classify data samples based on their estimated latent variables (low-dimensional features).

Natural ambiguity in the model parameterization translated into ambiguity in the direction of weight vector, $\mathbf{w}$. In other words, one can estimate the norm of $\mathbf{w}$ with arbitrary direction. It was shown that this simplification can not negatively affect likelihood function or classification rule, but it removes unnecessary parameter redundancy.

In MFM framework, the additional assumption on distribution of latent variables for each class will negatively affect data modeling and hence reconstruction quality compared to its counterpart in MPPCA. However, this assumption makes it possible to generate discriminative features which are suitable for classification in low-dimensional domain.

CHAPTER 4

PARAMETER ESTIMATION AND IMPLEMENTATION

4.1 Introduction

In this chapter, a supervised training based on the Expectation-Maximization (EM) and steepest descent algorithms is introduced to derive the Maximum Likelihood (ML) estimates of the unknown parameters of the model discussed in Chapter 3. It is shown that parameters associated with dimensionality reduction can be estimated using the EM algorithm whereas those of the classification system are estimated using a steepest descent algorithm.

The introduction of latent variables not only helped representing the data distribution and reducing the dimensionality in Chapter 3 but also in parameter estimation using EM algorithm which is used to find ML estimates of the parameters when the available data is incomplete. To accomplish this, we need to find explicit formulation for the squashed multivariate normal distribution which encompasses general cases of multivariate truncated normal [34] and skew-normal [35] distributions. Finding statistics of this random variable helped us finding analytic solution for EM algorithm in parameter estimation. Distribution of the latent variables for each class is also squashed multivariate normal which accommodates separability of the estimated features. For linear squashing boundary, we found the first and second order moments of this random variable analytically and used it in EM algorithm.

The outline of this chapter is as follows: Section 4.2 develops EM algorithm to find ML estimates of the model parameters for dimensionality reduction. Section 4.3 then provides steepest descent algorithm to estimate the classifier's parameters. Concluding remarks are given in Section 4.4.

4.2 EM Algorithm

Supervised learning in our method involves estimating matrices $\mathbf{G}_l$, mean vectors $\boldsymbol{\mu}_l$, variances σ_l^2, priors π_l, weight vector $\mathbf{w}$ and bias b using ML estimation. If we define the set of unknown parameters by $\boldsymbol{\Theta} \triangleq \{\{\boldsymbol{\Theta}_l\}_{l=1}^L, \|\mathbf{w}\|, b\}$, where $\boldsymbol{\Theta}_l \triangleq \{\mathbf{G}_l, \boldsymbol{\mu}_l, \pi_l, \sigma_l^2\}$, then given the i.i.d labeled training dataset $\mathcal{D} = \{\mathbf{x}_n, z_n\}_{n=1}^N$, we maximize the log-likelihood function

$$l(\boldsymbol{\Theta}|\mathcal{D}) = \sum_{n=1}^{N} \log f_{\boldsymbol{\Theta}}(\mathbf{x}_n, z_n)$$

with respect to all the unknown parameters, under $\sum_{l=1}^L \pi_l = 1$ constraint. We show that dimensionality reduction parameters, $\{\boldsymbol{\Theta}_l\}_{l=1}^L$, can be estimated using the EM algorithm.

Remark 4.1:

For the special case of $L = 1$ (i.e. no mixture), the likelihood function can be written as (refer to (3.26))

$$\begin{aligned} L(\boldsymbol{\Theta}|\mathcal{D}) &= \prod_{n=1}^{N} f_{\boldsymbol{\Theta}}(\mathbf{x}_n, z_n) \\ &= \prod_{n=1}^{N} f(\mathbf{x}_n) \Phi\Big(\frac{\mathbf{w}^T \hat{\mathbf{y}}_n + b}{\sqrt{1 + \mathbf{w}^T \mathbf{R}_{y|x} \mathbf{w}}}\Big)^{z_n} \Big(1 - \Phi\Big(\frac{\mathbf{w}^T \hat{\mathbf{y}}_n + b}{\sqrt{1 + \mathbf{w}^T \mathbf{R}_{y|x} \mathbf{w}}}\Big)\Big)^{1-z_n} \\ &= L_G \times L_D \end{aligned}$$

where $L_G = \prod_{n=1}^N f(\mathbf{x}_n)$ and $L_D = \prod_{n=1}^N \Phi\Big(\frac{\mathbf{w}^T \hat{\mathbf{y}}_n + b}{\sqrt{1+\mathbf{w}^T \mathbf{R}_{y|x} \mathbf{w}}}\Big)^{z_n} \Big(1 - \Phi\Big(\frac{\mathbf{w}^T \hat{\mathbf{y}}_n + b}{\sqrt{1+\mathbf{w}^T \mathbf{R}_{y|x} \mathbf{w}}}\Big)\Big)^{1-z_n}$. As it is shown in (2.16), L_G is the likelihood function of PPCA algorithm. Maximizing this likelihood function will guarantee that the estimated features are "generative", i.e., we can reconstruct the data samples $\mathbf{x}_n$ in the LS sense. On the other hand, maximizing L_D component alone, will guarantee that the estimated features $\hat{\mathbf{y}}_n$ are "discriminative", since the following conditions must be satisfied

$$\mathbf{w}^t \hat{\mathbf{y}}_n + b \gg 0 \text{ for } z = 1$$

$$\mathbf{w}^t \hat{\mathbf{y}}_n + b \ll 0 \text{ for } z = 0$$

to maximize L_D. That is, the estimated features must be linearly separable in $\mathbb{R}^d$ space via $\mathbf{w}^t\mathbf{y} + b$ discriminant function. However, the objective function in the proposed method is $L_G \times L_D$, which tries to find the parameters that can be used to estimate "generative-discriminative" features $\hat{\mathbf{y}}_n$. The same argument can be made for each component (or linear model) in $L > 1$ case as well though there is no direct relationship between the estimated features and likelihood function.

Since EM algorithm is used to find ML estimates of the parameters when the available data is incomplete [36], the introduction of latent variables helps in parameter estimation. In this model, the complete data samples, which cannot be fully observed, is denoted by $(\mathbf{x}, \mathbf{y}, z, l)$. Therefore, the complete log-likelihood function is

$$L(\mathbf{\Theta}) = \sum_{n=1}^{N} \log f_{\mathbf{\Theta}}(\mathbf{x}_n, \mathbf{y}_n, z_n, l_n).$$

However, the observable samples are only $(\mathbf{x}, z)$ and the pdf of incomplete data is given by

$$f_{\mathbf{\Theta}}(\mathbf{x}, z) = \sum_{l=1}^{L} \pi_l \int f_{\mathbf{\Theta}}(\mathbf{x}, \mathbf{y}, z, l) d\mathbf{y}.$$

EM algorithm maximizes the expected value of the complete log-likelihood function, conditioned on observed data and current iteration estimate of $\mathbf{\Theta}$ in two steps, Expectation and Maximization. It is shown that the successive estimates $\mathbf{\Theta}$ never decrease the likelihood function in each iteration. The two steps of the algorithm are:

1. **Expectation or E-Step**: At $(k+1)$-th iteration where the previous estimate of the parameters, $\hat{\mathbf{\Theta}}^k$ is available, calculate the expected value of complete log-likelihood function given the observable variables (training data) as a function of the parameters, $\mathbf{\Theta}$.

$$\begin{aligned} Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}^k) &= E[L(\mathbf{\Theta})|\mathcal{D}; \hat{\mathbf{\Theta}}^k] \\ &= \sum_{n=1}^{N} E[\log f_{\mathbf{\Theta}}(\mathbf{x}_n, \mathbf{y}_n, z_n, l_n)|\mathbf{x}_n, z_n; \hat{\mathbf{\Theta}}^k] \\ &= \sum_{n=1}^{N} \sum_{l=1}^{L} \int \log f_{\mathbf{\Theta}}(\mathbf{x}_n, \mathbf{y}, z_n, l) f(\mathbf{y}, l|\mathbf{x}_n, z_n; \hat{\mathbf{\Theta}}^k) d\mathbf{y} \end{aligned}$$

Note that the marginalization is done w.r.t the latent variables $\mathbf{y}$ and l by integral and summation, respectively, over all their possible values. Since $\mathbf{y}$ and l are dummy variables and $(\mathbf{x}, \mathbf{y}, z, l)$ samples are i.i.d, we can drop subscript n. Further, using the chain rule,

$$f(\mathbf{y}, l|\mathbf{x}_n, z_n; \hat{\mathbf{\Theta}}^k) = f_{\hat{\mathbf{\Theta}}^k}(l|\mathbf{x}_n, z_n) f(\mathbf{y}|\mathbf{x}_n, z_n, l; \hat{\mathbf{\Theta}}^k)$$

leads to simplified $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}^k)$

$$Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}^k) = \sum_{n=1}^{N} \sum_{l=1}^{L} f_{\hat{\mathbf{\Theta}}^k}(l|\mathbf{x}_n, z_n) E[\log f_{\mathbf{\Theta}}(\mathbf{x}_n, \mathbf{y}, z_n, l)|\mathbf{x}_n, z_n, l; \hat{\mathbf{\Theta}}^k]. \tag{4.1}$$

From now on we drop subscript $\mathbf{\Theta}$ for convenience. Using chain rule, $f(\mathbf{x}_n, \mathbf{y}, z_n, l)$ can be written as

$$f(\mathbf{x}_n, \mathbf{y}, z_n, l) = \pi_l f(\mathbf{y}) f(\mathbf{x}_n|\mathbf{y}, l) f(z_n|\mathbf{y}). \tag{4.2}$$

Note that $f(z|\mathbf{x}, \mathbf{y}, l) = f(z|\mathbf{y})$, since z given $\mathbf{y}$ is independent of $\mathbf{x}$ and model index, l, and $\mathbf{y}$ are independent as well. Taking logarithm from each component,

$$\log f(\mathbf{y}) = -\frac{d}{2} \log 2\pi - \frac{1}{2} \mathrm{tr}\{\mathbf{y}\mathbf{y}^T\}$$

$$\log f(\mathbf{x}_n|\mathbf{y}, l) = -\frac{D}{2} \log 2\pi - \frac{D}{2} \log \sigma_l^2 - \frac{1}{2\sigma_l^2} \mathrm{tr}\{(\mathbf{x}_n - \mathbf{G}_l\mathbf{y} - \boldsymbol{\mu}_l)(\mathbf{x}_n - \mathbf{G}_l\mathbf{y} - \boldsymbol{\mu}_l)^T\}$$

$$\log f(z_n|\mathbf{y}) = z_n \log \psi(g(\mathbf{y})) + (1 - z_n) \log (1 - \psi(g(\mathbf{y})))$$

$\log f(\mathbf{x}_n, \mathbf{y}, z_n, l)$ can be written as

$$\log f(\mathbf{x}_n, \mathbf{y}, z_n, l) = C + \log \pi_l - \frac{D}{2} \log \sigma_l^2 - \frac{1}{2\sigma_l^2} \mathrm{tr}\{(\mathbf{x}_n - \mathbf{G}_l\mathbf{y} - \boldsymbol{\mu}_l)(\mathbf{x}_n - \mathbf{G}_l\mathbf{y} - \boldsymbol{\mu}_l)^T\} \tag{4.3}$$

where C contains all other terms that are not dependent on the parameters $\mathbf{\Theta}_l$. Given the expression for $\log f(\mathbf{x}_n, \mathbf{y}, z_n, l)$ in (4.3), simplified $Q(\mathbf{\Theta}, \hat{\mathbf{\Theta}}^k)$ in (4.1) requires finding the posterior probabilities $p_{n,l} = f_{\hat{\mathbf{\Theta}}^k}(l|\mathbf{x}_n, z_n)$ and $\langle \mathbf{y} \rangle_{n,l} = E[\mathbf{y}|\mathbf{x}, z, l; \hat{\mathbf{\Theta}}^k]$ and $\langle \mathbf{y}\mathbf{y}^T \rangle_{n,l} = E[\mathbf{y}\mathbf{y}^T|\mathbf{x}, z, l; \hat{\mathbf{\Theta}}^k]$ evaluating with current iteration estimate of the parameters, $\hat{\mathbf{\Theta}}^k$. In the training phase, labeled data samples are assigned to the linear model,

l, according to the responsibility of l-th model, $p_{n,l}$, for generation of the n-th data sample which can be calculated by using Bayes' rule,

$$f(l|\mathbf{x}_n, z_n) = \frac{\pi_l f(\mathbf{x}_n, z_n|l)}{f(\mathbf{x}_n, z_n)} \tag{4.4}$$

where $f(\mathbf{x}, z)$ and $f(\mathbf{x}, z|l)$ were given in (3.23) and (3.26). Using the definition of conditional probability distribution and chain rule, we can find the distribution of the latent variable $\mathbf{y}$ given training data sample $(\mathbf{x}, z)$ and model index l as follows

$$\begin{aligned} f(\mathbf{y}|z, \mathbf{x}, l) &= \frac{f(\mathbf{x}, \mathbf{y}, z|l)}{f(\mathbf{x}, z|l)} \\ &= \frac{f(\mathbf{x}|l) f(\mathbf{y}|\mathbf{x}, l) f(z|\mathbf{y})}{f(\mathbf{x}|l) f(z|\mathbf{x}, l)} \\ &= f(\mathbf{y}|\mathbf{x}, l) \frac{f(z|\mathbf{y})}{f(z|\mathbf{x}, l)} \\ &= f(\mathbf{y}|\mathbf{x}, l) (\frac{\Phi(\mathbf{w}^T\mathbf{y} + b)}{E[\Phi(\mathbf{w}^T\mathbf{y} + b)|\mathbf{x}, l]})^z (\frac{1 - \Phi(\mathbf{w}^T\mathbf{y} + b)}{1 - E[\Phi(\mathbf{w}^T\mathbf{y} + b)|\mathbf{x}, l]})^{1-z} \end{aligned}$$

which is squashed multivariate normal for both $z = 0$ and $z = 1$ cases. Therefore, the first and second order moments of $\mathbf{y}$ given $(\mathbf{x}, z, l)$ can be summarized as follows,

$$E[\mathbf{y}|\mathbf{x}_n, z_n, l] = \boldsymbol{\mu}_{y|x_n,l} + \tilde{\boldsymbol{\mu}}_{x_n,z_n,l} \tag{4.5}$$

$$E[\mathbf{y}\mathbf{y}^T|\mathbf{x}_n, z_n, l] = \mathbf{R}_{y|x,l} + \boldsymbol{\mu}_{y|x_n,l}\boldsymbol{\mu}_{y|x_n,l}{}^T + \tilde{\boldsymbol{\mu}}_{x_n,z_n,l}\boldsymbol{\mu}_{y|x_n,l}^T + \boldsymbol{\mu}_{y|x_n,l}\tilde{\boldsymbol{\mu}}_{x_n,z_n,l}^T + \tilde{\mathbf{R}}_{x_n,z_n,l} \tag{4.6}$$

where

$$\tilde{\boldsymbol{\mu}}_{x_n,z_n,l} = \gamma(\mathbf{x}_n, z, l)\mathbf{R}_{y|x,l}\mathbf{w}, \tag{4.7}$$

$$\tilde{\mathbf{R}}_{x_n,z_n,l} = -\gamma(\mathbf{x}_n, z, l)\frac{\mathbf{w}^T\boldsymbol{\mu}_{y|x_n,l} + b}{1 + \mathbf{w}^T\mathbf{R}_{y|x,l}\mathbf{w}}\mathbf{R}_{y|x,l}\mathbf{w}\mathbf{w}^T\mathbf{R}_{y|x,l}, \tag{4.8}$$

$$\gamma(\mathbf{x}_n, z, l) = \frac{\phi\Big(\frac{\mathbf{w}^T\boldsymbol{\mu}_{y|x_n,l}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}_{y|x,l}\mathbf{w}}}\Big)}{\sqrt{1 + \mathbf{w}^T\mathbf{R}_{y|x,l}\mathbf{w}}\Big(\Phi\Big(\frac{\mathbf{w}^T\boldsymbol{\mu}_{y|x_n,l}+b}{\sqrt{1+w^T\mathbf{R}_{y|x,l}\mathbf{w}}}\Big) - \mathbf{1}_{\{0\}}(z_n)\Big)}. \tag{4.9}$$

which can be found by merging the results of squashed multivariate normal distribution of both $z = 1$ and $z = 0$ cases and replacing posterior mean, $\boldsymbol{\mu}_{y|x,1}$, and covariance, $\mathbf{R}_{y|x,l}$, in the moments calculated in Appendix A.

2. **Maximization or M-Step**: Update the estimate of parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ by maximizing $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}^k)$ w.r.t the parameter of interest, $\boldsymbol{\theta}$, i.e.

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}^k) &= \sum_{n=1}^{N} \sum_{l=1}^{L} p_{n,l} \nabla_{\boldsymbol{\theta}} E[\log f(\mathbf{x}_n, \mathbf{y}, z_n, l) | \mathbf{x}_n, z_n, l; \hat{\boldsymbol{\Theta}}^k] \\ &= 0. \end{aligned} \tag{4.10}$$

If we replace (4.3) in (4.10) and take gradient w.r.t the model parameters and apply $\sum_{l=1}^{L} \pi_l = 1$ constraint, we have

$$N_l = \sum_{n=1}^{N} p_{n,l}$$

$$\pi_l = \frac{N_l}{N}$$

$$\boldsymbol{\mu}_l = \frac{1}{N_l} \sum_{n=1}^{N} p_{n,l} (\mathbf{x}_n - \mathbf{G}_l \langle \mathbf{y} \rangle_{n,l})$$

$$\mathbf{G}_l = \Big(\sum_{n=1}^{N} p_{n,l} (\mathbf{x}_n - \boldsymbol{\mu}_l) \langle \mathbf{y} \rangle_{n,l} \Big) \Big(\sum_{n=1}^{N} p_{n,l} \langle \mathbf{y}\mathbf{y}^T \rangle_{n,l} \Big)^{-1}$$

$$\sigma_l^2 = \frac{1}{N_l D} \sum_{n=1}^{N} p_{n,l} \Big(\|\mathbf{x}_n - \boldsymbol{\mu}_l\|^2 - 2(\mathbf{x}_n - \boldsymbol{\mu}_l)^T \mathbf{G}_l \langle \mathbf{y} \rangle_{n,l} + \text{tr } \mathbf{G}_l^T \mathbf{G}_l \langle \mathbf{y}\mathbf{y}^T \rangle_{n,l} \Big)$$

Update equations for $\boldsymbol{\mu}_l$ and $\mathbf{G}_l$ are coupled and some simplifications are needed to decouple them. The simplified decoupled versions of EM implementation are provided in Table 4.1.

Table 4.1: Steps of the EM Algorithm

Require: L and the number of iterations, I.

Input: The labeled training dataset $\mathcal{D} = \{\mathbf{x}_n, z_n\}_{n=1}^N$ and $\mathbf{w}$, b.

Output: ML estimate of the parameters $\{\mathbf{G}_l, \boldsymbol{\mu}_l, \pi_l, \sigma_l^2\}_{l=1}^L$.

Initialization Initialize $\mathbf{G}_l$ and $\boldsymbol{\mu}_l$ randomly, priors $\pi_l = \frac{1}{L}$ and choose large variances σ_l^2 or use pre-determined values. Set the iteration index, $i = 1$.

for $i \leq I$ **do**

***Expectation Step*:** Given the current parameters, evaluate the following estimates.

for all l **do**

$N_l = \sum_{n=1}^N p_{n,l}$

$\mathbf{m}_{\mathbf{x}|l} = \frac{1}{N_l} \sum_{n=1}^N p_{n,l} \mathbf{x}_n$

$\mathbf{m}_{\mathbf{y}|l} = \frac{1}{N_l} \sum_{n=1}^N p_{n,l} \langle \mathbf{y} \rangle_{n,l}$

$\mathbf{R}_{\mathbf{yy}|l} = \frac{1}{N_l} \sum_{n=1}^N p_{n,l} \langle \mathbf{y}\mathbf{y}^T \rangle_{n,l}$

$\mathbf{R}_{\mathbf{xy}|l} = \frac{1}{N_l} \sum_{n=1}^N p_{n,l} \mathbf{x}_n \langle \mathbf{y} \rangle_{n,l}^T$

$\mathbf{R}_{\mathbf{xx}|l} = \frac{1}{N_l} \sum_{n=1}^N p_{n,l} \mathbf{x}_n \mathbf{x}_n^T$

$\mathbf{C}_{\mathbf{yy}|l} = \mathbf{R}_{\mathbf{yy}|l} - \mathbf{m}_{\mathbf{y}|l} \mathbf{m}_{\mathbf{y}|l}^T$

$\mathbf{C}_{\mathbf{xy}|l} = \mathbf{R}_{\mathbf{xy}|l} - \mathbf{m}_{\mathbf{x}|l} \mathbf{m}_{\mathbf{y}|l}^T$

end for

where $p_{n,l} = f(l|\mathbf{x}_n, z_n)$, $\langle \mathbf{y} \rangle_{n,l} = E[\mathbf{y}|\mathbf{x}_n, z_n, l]$ and $\langle \mathbf{y}\mathbf{y}^T \rangle_{n,l} = E[\mathbf{y}\mathbf{y}^T|\mathbf{x}_n, z_n, l]$. The expressions of these terms can be found in (4.4), (4.5) and (4.6), respectively.

***Maximization Step*:** Update the current parameters.

for all l **do**

$\pi_l = \frac{N_l}{N}$

$\boldsymbol{\mu}_l = \mathbf{m}_{\mathbf{x}|l} - \frac{1}{1 - \mathbf{m}_{\mathbf{y}|l}^T \mathbf{R}_{\mathbf{yy}|l}^{-1} \mathbf{m}_{\mathbf{y}|l}} \mathbf{C}_{\mathbf{xy}|l} \mathbf{R}_{\mathbf{yy}|l}^{-1} \mathbf{m}_{\mathbf{y}|l}$

$\mathbf{G}_l = \mathbf{C}_{\mathbf{xy}|l} \mathbf{C}_{\mathbf{yy}|l}^{-1}$

$\sigma_l^2 = \frac{1}{D} \Big(\mathrm{tr}\{\mathbf{R}_{\mathbf{xx}|l}\} + \boldsymbol{\mu}_l^T \boldsymbol{\mu}_l - 2\mathrm{tr}\{\mathbf{G}_l^T \mathbf{R}_{\mathbf{xy}|l}\} + 2\boldsymbol{\mu}_l^T (\mathbf{G}_l \mathbf{m}_{\mathbf{y}|l} - \mathbf{m}_{\mathbf{x}|l}) + \mathrm{tr}\{\mathbf{G}_l^T \mathbf{G}_l \mathbf{R}_{\mathbf{yy}|l}\} \Big)$

end for

$i := i + 1$.

end for

4.3 Steepest Descent Algorithm

As mentioned before, to estimate the the classification parameters, $\{\|\mathbf{w}\|, b\}$, we use steepest descent algorithm,

$$\theta^{k+1} = \theta^k + \alpha \nabla_\theta l(\mathbf{\Theta}^k|\mathcal{D}) \tag{4.11}$$

where $\nabla_\theta l(\mathbf{\Theta}^k|\mathcal{D})$ is the gradient of the log-likelihood function w.r.t parameters $\theta \in \{\|\mathbf{w}\|, b\}$,

$$\nabla_\theta l(\mathbf{\Theta}|\mathcal{D}) = \sum_{n=1}^{N} \frac{1}{f_{\mathbf{\Theta}}(\mathbf{x}_n, z_n)} \nabla_\theta f_{\mathbf{\Theta}}(\mathbf{x}_n, z_n)$$

and α is the learning rate.

The update equations for b and $\|\mathbf{w}\|$ can be obtained by taking gradient of (3.20) w.r.t these parameters. The summary of implementation of steepest descent algorithm for estimating $\|\mathbf{w}\|$ and b in training phase is provided in Table 4.2.

Table 4.2: Steps of the Steepest Descent Algorithm

Require: Learning rates α_w, α_b and number of iterations, I.

Input: The labeled training dataset $\mathcal{D} = \{\mathbf{x}_n, z_n\}_{n=1}^N$ and $\{\mathbf{G}_l, \boldsymbol{\mu}_l, \pi_l, \sigma_l^2\}_{l=1}^L$.

Output: ML estimate of the parameters $\|\mathbf{w}\|_2$ and b.

Initialization Initialize $\mathbf{w} = \frac{1}{\sqrt{d}}\mathbf{1}$ and $b = 0$ or use pre-determined values, and set the iteration index, $i = 1$.

for $i \leq I$ **do**

$$\|\mathbf{w}\|^{k+1} = \|\mathbf{w}\|^k - \alpha_w \sum_{n=1}^{N} \sum_{l=1}^{L} \frac{(-1)^{z_n} \pi_l f(\mathbf{x}_n|l)}{f(\mathbf{x}_n, z_n)} \frac{\phi\Big(\frac{\mathbf{w}^T \boldsymbol{\mu}_{y|x_n,l} + b}{\sqrt{1+\mathbf{w}^T \mathbf{R}_{y|x,l}\mathbf{w}}}\Big)}{\|\mathbf{w}\|(1+\mathbf{w}^T \mathbf{R}_{y|x,l}\mathbf{w})^{\frac{3}{2}}} (\mathbf{w}^T \boldsymbol{\mu}_{y|x_n,l} - \mathbf{w}^T \mathbf{R}_{y|x,l}\mathbf{w}b)|_{\mathbf{w}=\mathbf{w}^k}$$

$$b^{k+1} = b^k - \alpha_b \sum_{n=1}^{N} \frac{(-1)^{z_n}}{f(\mathbf{x}_n, z_n)} \sum_{l=1}^{L} \pi_l f(\mathbf{x}_n|l) \frac{\phi\Big(\frac{\mathbf{w}^T \boldsymbol{\mu}_{y|x_n,l} + b}{\sqrt{1+\mathbf{w}^T \mathbf{R}_{y|x,l}\mathbf{w}}}\Big)}{\sqrt{1+\mathbf{w}^T \mathbf{R}_{y|x,l}\mathbf{w}}}|_{\mathbf{w}=\mathbf{w}^{k+1}, b=b^k}$$

$i := i + 1$

end for

Table 4.3 demonstrates the full implementation of the proposed algorithm, including training phase and testing for new data. In training the parameters are estimates whereas in testing phase, dimensionality reduction, clustering, reconstruction and classification are performed on a new data.

Table 4.3: Summary of Implementation Steps

Require: d, L, the number of iterations, I_1 and I_2 and *threshold*.

Input: The labeled training dataset $\mathcal{D} = \{\mathbf{x}_n, z_n\}_{n=1}^N$ and a new unknown sample $\mathbf{x}_{new}$.

Output: ML estimate of the parameters, estimated low-dimensional feature $\hat{\mathbf{y}}_{new}$, estimated label $\hat{z}_{new}$, model index $\hat{l}_{new}$ and reconstructed sample $\hat{\mathbf{x}}_{new}$.

Initialization Initialize log-likelihood, $\mathcal{L} = -\infty$ and $error = +\infty$

while $error > thershold$ **do**

Find ML estimate of the parameters $\{\mathbf{G}_l, \boldsymbol{\mu}_l, \pi_l, \sigma_l^2\}_{l=1}^L$ via EM algorithm (I_1 iterations).

Find ML estimate of the parameters $\|\mathbf{w}\|_2$ and b via steepest descent method. (I_2 iterations)

Calculate log-likelihood of the parameters:

$$\mathcal{L}^* = \sum_{n=1}^{N} \log f_{\boldsymbol{\Theta}}(\mathbf{x}_n, z_n).$$

Update the *error*:

$$error = |\frac{\mathcal{L} - \mathcal{L}^*}{\mathcal{L}}|$$

Update the log-likelihood: $\mathcal{L} = \mathcal{L}^*$.

end while

Find $\hat{\mathbf{y}}_{new}$, $\hat{z}_{new}$, $\hat{l}_{new}$ and $\hat{\mathbf{x}}_{new}$ using (3.6), (3.19), (3.15), and (3.16) .

4.4 Conclusion

In this chapter, a combined EM and steepest descent algorithm is introduced to derive the ML estimates of the unknown parameters of the model. It was shown that parameters associated with dimensionality reduction can be estimated using the EM algorithm whereas those of the classification system are estimated using steepest descent algorithm. Squashed multivariate normal distribution and its moments, derived in Appendix A, helped us finding analytical and explicit solutions in the E-Step. It must be mentioned that these algorithms are *batch* and *iterative* in nature, i.e. they need all the training data in each iteration to estimate the parameters. However, by interpreting EM as an alternating maximization of a negative free-energy-like function [37], it is possible to derive online EM algorithms, suitable for online learning (e.g. in situations where the data arrives one at a time). Additionally,

EM has slow convergence after the first few steps. Despite these shortcomings, EM usually remains the best choice for parameter estimation when there are some unobservable (latent) variables. Steepest descent's issue is the learning rate that needs to be selected using a trial and error procedure. These algorithms are used in the Chapter 5 to estimate the model parameters to perform dimensionality reduction, clustering, reconstruction, and classification.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Introduction

In this chapter, the developed *Mixture of Factor Models* (MFM) is applied to two datasets to assess and compare its performance in dimensionality reduction, clustering, reconstruction, and classification. In the first section, we introduce different measures to quantify performance of each task. Sample average of the dimensionality reduction and clustering risks are the only measures to evaluate the dimensionality reduction and clustering algorithms. In order to evaluate the overall analysis-synthesis reconstruction performance, we use the signal-to-error ratio (SER) measure of the reconstruction process. These measures are common among PPCA and MPPCA algorithms and have been used for the purpose of comparison. On the other hand, empirical probability of correct classification is a useful measure that compares the classification performance of the proposed method with any other methods, such as SVM. However, sample average of classification risk gives us a better and more accurate measure of confidence of correct classification, which is used for MFM.

This chapter describes the experiments conducted on two sets of image databases, namely Synthetic Aperture Sonar (SAS) and facial images. The first dataset consists of 181 SAS images of model-generated underwater objects. The second dataset consists of 186 facial images of different individuals. The goal is to compare all the aforementioned functionalities of the proposed MFM method with those of the corresponding methods using these measures.

The organization of this chapter is as follows: Section 5.2 provides different measures used in this chapter to quantify performance of dimensionality reduction, clustering, reconstruction and classification tasks. Sections 5.3 and 5.4 provide experimental results on the SAS and facial image data sets, respectively. Finally, concluding remarks are given in Section 5.5.

5.2 Adopted Performance Measures

The proposed MFM method offers a unified framework to analyze high-dimensional data. More specifically, its clustering, dimensionality reduction and reconstruction performance is compared against those of PPCA and MPPCA which are motivated by the objective of deriving a reduced-dimensionality representation of the data, with minimum reconstruction error. Sample average of the dimensionality reduction risk per dimension,

$$\hat{R}_{dim.} = \frac{1}{Nd} \sum_{n=1}^{N} R^*_{dim.}(\mathbf{x}_n) \tag{5.1}$$

and sample average of the clustering risk,

$$\hat{R}_{clus.} = \frac{1}{N} \sum_{n=1}^{N} R^*_{clus.}(\mathbf{x}_n) \tag{5.2}$$

are appropriate measures to assess and compare the dimensionality reduction and clustering performance. In order to quantify the overall analysis-synthesis performance, or simply the reconstruction quality, we use the SER defined as

$$\text{SER} = 20 \log_{10} \frac{1}{N} \sum_{n=1}^{N} \frac{\|\mathbf{x}_n\|_2}{\|\hat{\mathbf{x}}_n - \mathbf{x}_n\|_2}. \tag{5.3}$$

On the other hand, we compare classification performance with SVM using the empirical probability of correct classification

$$\hat{P}_{cc} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}_{z_n}(\hat{z}_n) \tag{5.4}$$

which is the most popular measure of classification performance. For MFM, label variables can be estimated based on data samples, $\mathbf{x}_n$ or alternatively the associated low-dimensional features, $\hat{\mathbf{y}}_n$. Empirical probability of correct classification might be misleading when the number of test samples are small. The alternative measures to assess classification confidence are the sample average of classification risk based on data samples, $\mathbf{x}_n$,

$$\hat{R}^{\mathbf{x}}_{clas.} = \frac{1}{N} \sum_{n=1}^{N} R^*_{clas.}(\mathbf{x}_n) \tag{5.5}$$

or based on the estimated low-dimensional features $\hat{\mathbf{y}}_n$

$$\hat{R}^{\mathbf{y}}_{clas.} = \frac{1}{N}\sum_{n=1}^{N} R^*_{clas.}(\hat{\mathbf{y}}_n) \tag{5.6}$$

which are unique to Bayesian classifiers.

Among the important theoretical results are asymptotic behaviors of dimensionality reduction, classification and reconstruction methods, which can be investigated in these experiments. The following table provides a summary of these asymptotic simplifications.

Table 5.1: Asymptotic behavior of dimensionality reduction, classification and reconstruction methods.

	$R^*_{dim.}(\mathbf{x}_n) \to 0$	$R^*_{clus.}(\mathbf{x}_n) \to 0$
$\hat{\mathbf{y}}_n$	-	$(\sigma^2_{\hat{l}_n}\mathbf{I} + \mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n})^{-1}\mathbf{G}^T_{\hat{l}_n}(\mathbf{x}_n - \boldsymbol{\mu}_{\hat{l}_n})$
$\hat{z}_n$	$\mathbf{1}_{(0,+\infty)}(\mathbf{w}^T\hat{\mathbf{y}}_n + b)$	$\mathbf{1}_{(0,+\infty)}(\mathbf{w}^T\hat{\mathbf{y}}_n + b)$
$\hat{\mathbf{x}}_n$	$\mathbf{G}_{\hat{l}_n}\hat{\mathbf{y}}_n + \boldsymbol{\mu}_{\hat{l}_n}$	$P_{\mathbf{G}_{\hat{l}_n}}(\mathbf{x}_n - \boldsymbol{\mu}_{\hat{l}_n}) + \boldsymbol{\mu}_{\hat{l}_n}$

The significance of having two different classification measures, based on data samples $\mathbf{x}_n$ as well as associated low-dimensional features $\hat{\mathbf{y}}_n$ is to approve the theoretical asymptotic behaviors of MFM.

5.3 Results on SAS Image Data Set

The image data set used in this section contains SAS images of model-generated objects superimposed on noisy backgrounds. The data set consists of high frequency (HF) sonar images which have high spatial resolution that provides the ability to capture different target characteristics. These images correspond to two different object types namely Cylinder (target or mine-like) and Block (non-target or non-mine-like). Each image snippet is originally of size 260×540 pixels. For computational convenience, these images are then sub-sampled to 65×135 images and vectorized to produce data vectors $\mathbf{x}_n$ with dimension $D = 8775$.

The signatures of each object are synthetically generated and are placed at various aspect angles from 1 to 180 degrees for each object type.

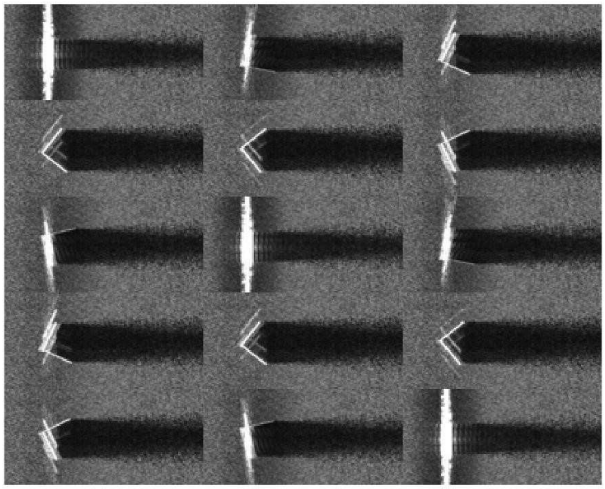

Figure 5.1: Samples of Block Snippets (Non-Target).

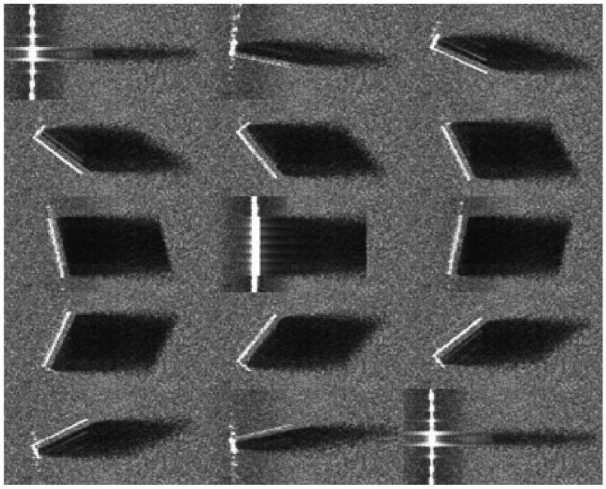

Figure 5.2: Samples of Cylinder Snippets (Target).

The goal of the classification is to assign non-target versus target labels to these image snippets. Figures 5.1 and 5.2 give 15 different examples of target and non-target snippets, respectively, which are uniformly selected from aspect angles $1 - 180$ degrees with 12 degree separation. Two-fold cross-validation method is used to evaluate the results, i.e. the training set consists of 50% randomly chosen image snippets at different aspects while using the rest as the test set and vice versa. The average results of the two experiments is reported as the final result. For our experiment, the dimension of the low-dimensional feature space is chosen to be $d = 3, 6, 9$ and number of linear model $L = 1, 2, 3$.

Table 5.2: Performance measures of the MFM method for different choices of d and L - SAS data set.

L	d	$\hat{R}_{dim.}$	$\hat{R}^{\mathbf{x}}_{clas.}$	$\hat{R}^{\mathbf{y}}_{clas.}$	$\hat{R}_{clus.}$	$\hat{P}^{\mathbf{x}}_{cc}$	$\hat{P}^{\mathbf{y}}_{cc}$	SER
1	3	0.0017	0.0045	0.0045	0	1	1	15.75
2	3	0.0013	0.0099	0.0099	2.8710e-14	1	1	17.39
3	3	0.0014	0.0042	0.0042	2.3086e-09	1	1	18.76
1	6	0.0063	0.0060	0.0060	0	1	1	17.65
2	6	0.0073	0.0094	0.0094	2.6755e-26	1	1	19.83
3	6	0.0047	0.0096	0.0096	1.3020e-64	1	1	21.42
1	9	0.0086	0.0058	0.0058	0	1	1	19.12
2	9	0.0066	0.0094	0.0093	9.1295e-73	1	1	21.66
3	9	0.0055	0.0096	0.0096	2.0454e-285	1	1	23.14

Table 5.2 provides the experimental results of applying MFM to SAS data set. Results are presented in terms of performance measures, e.g., sample average of dimensionality reduction, $\hat{R}_{dim.}$, and clustering, $\hat{R}_{clus.}$, risks, SER in dB, sample average of classification risk and probabilities of correct classification based on data samples $\mathbf{x}_n$, $\hat{R}^{\mathbf{x}}_{clas.}$ and $\hat{P}^{\mathbf{x}}_{cc}$, or based on estimated low-dimensional features $\hat{\mathbf{y}}_n$, $\hat{R}^{\mathbf{y}}_{clas.}$ and $\hat{P}^{\mathbf{y}}_{cc}$.

Table 5.3: Performance measures of the PPCA ($L = 1$) and MPPCA ($L > 1$) methods for different choices of d and L - SAS data set.

L	d	$\hat{R}_{dim.}$	$\hat{R}_{clus.}$	SER
1	3	3.0632e-04	0	15.75
2	3	0.0056	7.6220e-21	17.84
3	3	0.0034	1.1223e-110	19.65
1	6	3.6321e-04	0	17.63
2	6	0.0069	2.2151e-183	20.29
3	6	0.0052	5.0288e-16	22.24
1	9	4.2536e-04	0	19.12
2	9	0.0068	4.2120e-42	21.83
3	9	0.0060	4.9142e-264	23.63

Table 5.3, on the other hand, provides experimental results of applying PPCA ($L = 1$) and MPPCA ($L > 1$) on SAS data set. It provides the dimensionality reduction and clustering risks, and SER. There are two different clustering mechanisms i.e. hard vs. soft clustering [1]. In soft clustering, which is employed in MFM as well as MPPCA, one talks about the probability of a data sample belonging to a cluster. On the other hand, in hard clustering each cluster contains exclusive set of data samples. For both MFM and MPPCA methods, sample average of clustering risk (for $L > 1$), was insignificant. In other words, each data sample is assigned to a specific linear model with high probability. As mentioned before, this result mimics hard clustering method. However, since the objective in MPPCA is data modeling, irrespective of classification performance, it is expected to perform clustering with more confidence, which is also obvious in the results as sample average of clustering risk for MPPCA is smaller than of the MFM.

In general, as the number of latent variables increases the data model becomes more complex. This happens by either increasing dimension of continuous latent variables, $\mathbf{y}$, or the number of discrete latent variables, L. Sample average of dimensionality reduction risk is in direct relation with model complexity i.e. the more complex the model is, the less confidence are the estimates for latent variables. Other factors play important role as well. For instance, for $L > 1$ there are multiple local likelihood maxima that the EM can achieve, which will lead to different results. Data structure is the other factor. In general, as the estimated parameters and model complexity describe the data model better, it will achieve a smaller risk of dimensionality reduction.

When the clustering risk is negligible, it can be shown that (3.7) can be simplified to

$$R^*_{dim.}(\mathbf{x}_n) = \mathrm{tr}\{(\mathbf{I} + \frac{1}{\sigma^2_{\hat{l}_n}}\mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n})^{-1}\}. \tag{5.7}$$

which is not a function of data sample $\mathbf{x}_n$ any more. In this case, there are two major components affecting dimensionality reduction risk. The first component is the eigenvalues of $\frac{1}{\sigma^2_{\hat{l}_n}}\mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n}$. The larger these eigenvalues are, the better the estimated subspace $\mathbf{G}_{\hat{l}_n}$ models the training data samples hence leading to lower risk. The second component is the dimension of latent variables, d. As the dimension of latent variables increases, the risk of dimensionality reduction per dimension increases as well. This property can easily be shown for PPCA algorithm. In Chapter 2, it was shown that the columns of $\mathbf{G}$ are the principal eigenvalues of the data covariance matrix weighted by their corresponding eigenvalues (discarding noise variance). As we increase the dimension of the latent variable $\mathbf{y}$, we subsequently increase the dimension of $\mathbf{G}$. The added columns helps capturing more of the covariance structure of the data, majority of which was captured before. The reason is that the added dimensions are influenced by minor eigenvalues of the data covariance matrix, therefore the risk of the added dimension is more than the previous ones. This will lead to an increase in risk of dimensionality reduction per dimension. However, we can not make the same argument about MPPCA or MFM because these methods don't have *additivity* property (see Chapter 2).

For the same model complexity, i.e. same d and L, MPPCA and PPCA yield a much smaller dimensionality reduction risk than MFM. Since the average clustering risk is small, the only explainable factor is that the eigenvalues of $\frac{1}{\sigma^2_{\hat{l}_n}}\mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n}$ matrix for MPPCA/PPCA is larger than their counterparts in MFM. The reason is that maximizing the likelihood function in PPCA/MPPCA algorithm is equivalent to best modeling the data distribution, whereas in MFM likelihood maximization yields to a model for data and label. This additional constraint in MFM prevents it to model data distribution as well as PPCA/MPPCA. Therefore, the eigenvalues of $\frac{1}{\sigma^2_{\hat{l}_n}}\mathbf{G}^T_{\hat{l}_n}\mathbf{G}_{\hat{l}_n}$ matrix, which are indicative of a quality measure of data modeling, are larger for PPCA and MPPCA methods.

In order to quantify reconstruction quality, signal to reconstruction error ratio was used. For the linear models, $L = 1$, both MFM and PPCA yield the same reconstruction SER. The reason behind this equality is that original SAS images (target and non-target) were originally linearly separable. This can be explained by observing estimated latent variables, for $d = 3$ $L = 1$, in PPCA algorithm. Therefore, assuming linear separability of the latent variable, in MFM, doesn't impact the quality of data modeling detrimentally.

In order to evaluate the reconstruction performance visually and study the effect of d and model complexity L on reconstruction SER, we chose a random selection of target/non-target images that were mapped to the d-dimensional latent space and the reconstructed using the procedure in Chapter 3. For the $L = 1$ case, there were no tangible differences between MFM and PPCA results hence only MFM results are presented in Figures 5.4-5.6 (for $d = 3, 6, 9$). On the other hand, Figures 5.7-5.10 show the reconstructed SAS images using both MFM and MPPCA methods for $d = 3$ and $L = 2, 3$.

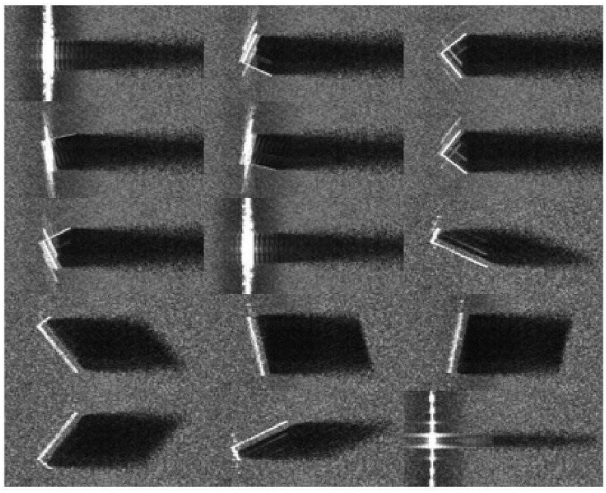

Figure 5.3: An specific selection of original target and non-target SAS images.

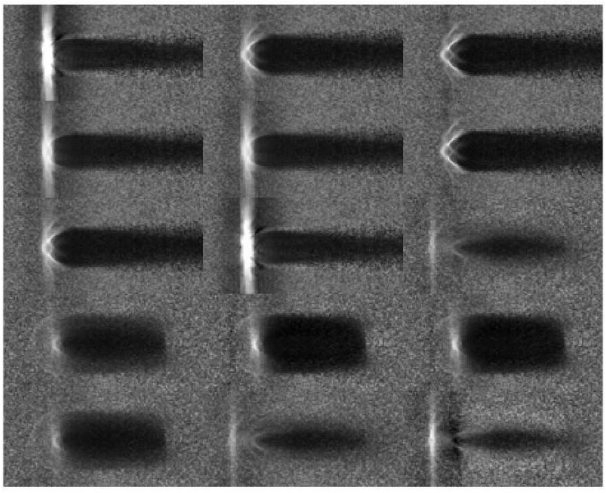

Figure 5.4: Reconstructed SAS images using FM method with $L = 1$ and $d = 3$.

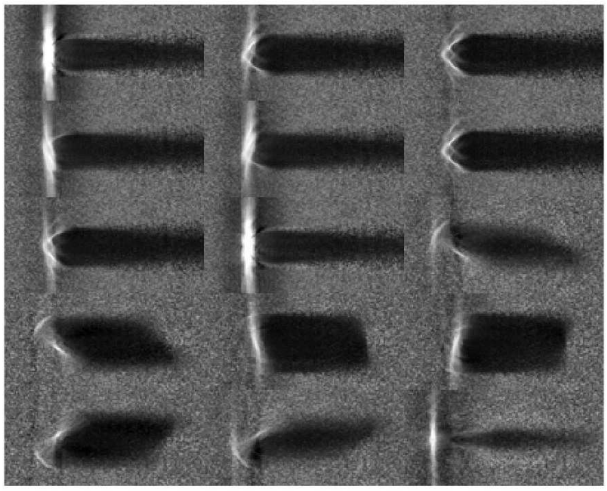

Figure 5.5: Reconstructed SAS images using FM method with $L = 1$ and $d = 6$.

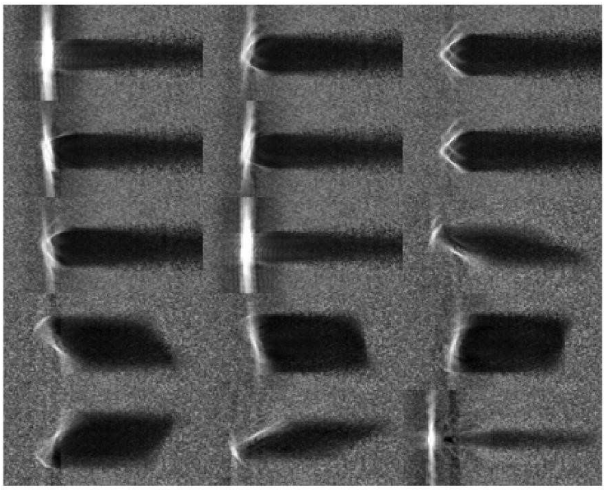

Figure 5.6: Reconstructed SAS images using FM method with $L = 1$ and $d = 9$.

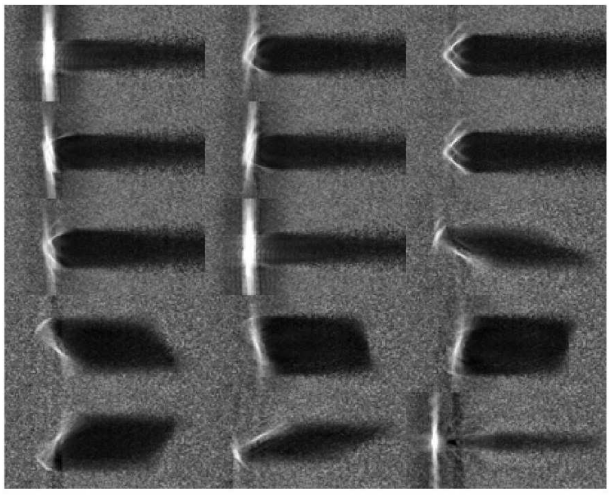

Figure 5.7: Reconstructed SAS images using MFM method with $L = 2$ and $d = 3$.

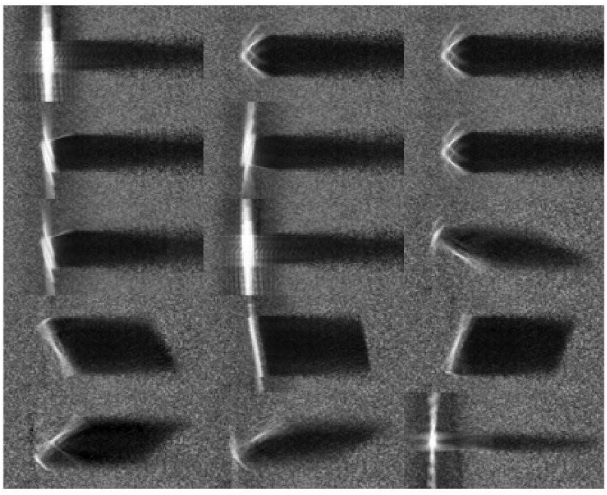

Figure 5.8: Reconstructed SAS images using MFM method with $L = 3$ and $d = 3$.

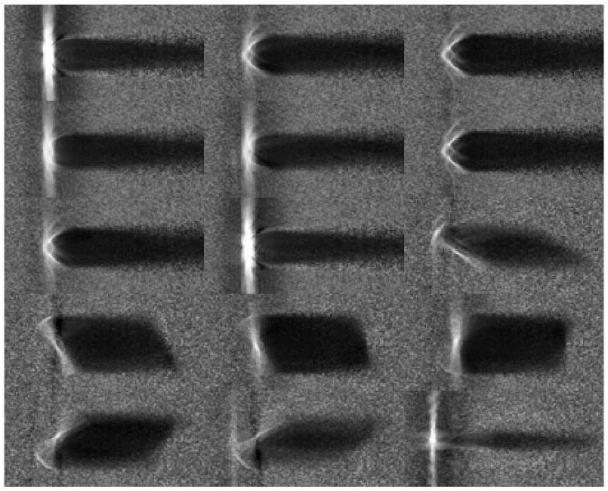

Figure 5.9: Reconstructed SAS images using MPPCA method with $L = 2$ and $d = 3$.

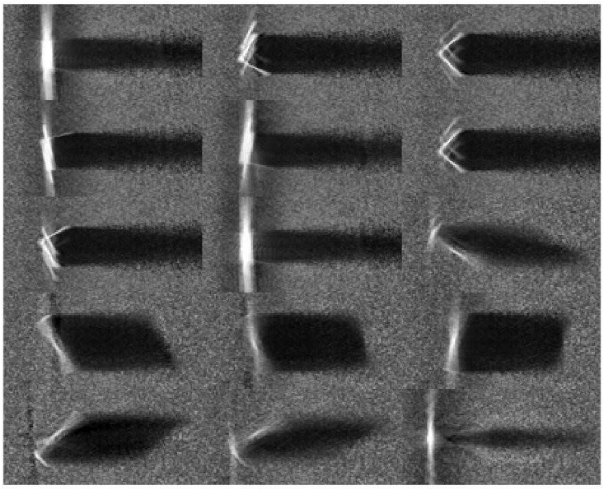

Figure 5.10: Reconstructed SAS images using MPPCA method with $L = 3$ and $d = 3$.

One of the obvious conclusions from these experiments is that increasing the dimension of latent variables, d, leads to a better reconstruction quality in terms of SER. As the number of columns of $\mathbf{G}$ increases, we have more flexibility to reconstruct image details (since the mean is already captured) as a linear combinations of them.

Increasing the model complexity by increasing the number of linear models will improve reconstruction performance as well. While the dimension of latent variables is fixed, adding linear models enables us cluster the data and reconstruct them with the best available local linear model. However, this clustering and reconstruction mechanism do not have the same degrees of freedom as linear model with the larger dimensions, thus it doesn't achieve the same reconstruction SER (compare cases where $d \times L$ are equal in Tables 5.3 as well as Table 5.2 to see the differences between local ($L = 1$)and global ($L > 1$) data modeling). For instance in $d = 3$ and $L = 2$ model, data set is clustered into two sets and reconstruction is done with 3 basis but in $d = 6$ and $L = 1$ case there is no clustering involved and the whole data is reconstructed using 6 basis which will obviously achieve a better reconstruction SER.

Generally, MPPCA leads to a better reconstructed images with higher reconstruction SER. However, there is only one visual superiority of the reconstructed images using MFM over MPPCA. Reconstructed highlights and shadows, which are among discriminative features of these SAS images, seem to be sharper than of those of the MPPCA.

The main distinction between MFM and MPPCA is the assumption of latent variable distribution for different classes with the premise of producing generative and discriminative features. It is worthy to compare distribution of estimated latent variables (or low-dimensional features) of MFM with PPCA/MPPCA in this experiment. Figures 5.11-5.16 provide estimated low dimensional features using both MPPCA/PPCA and MFM methods, for $d = 3$ and $L = 1, 2, 3$, respectively.

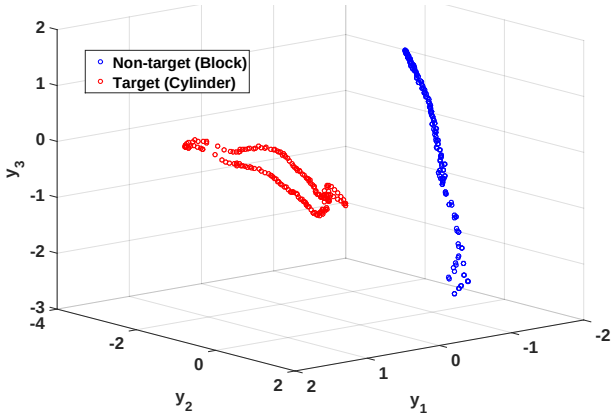

Figure 5.11: Distribution of estimated latent variables for SAS Images, PPCA method with $d = 3$.

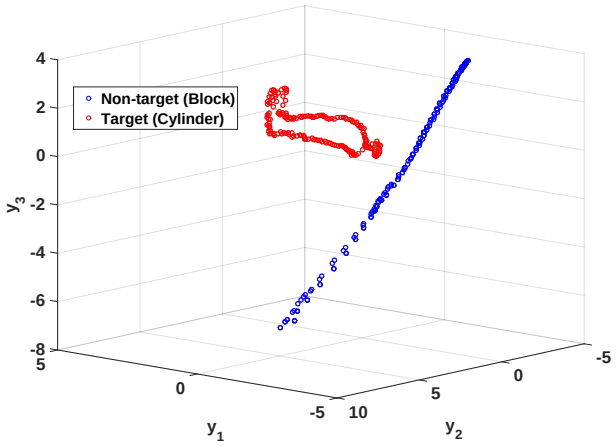

Figure 5.12: Distribution of estimated latent variables for SAS Images, FM method with $L = 1$ and $d = 3$.

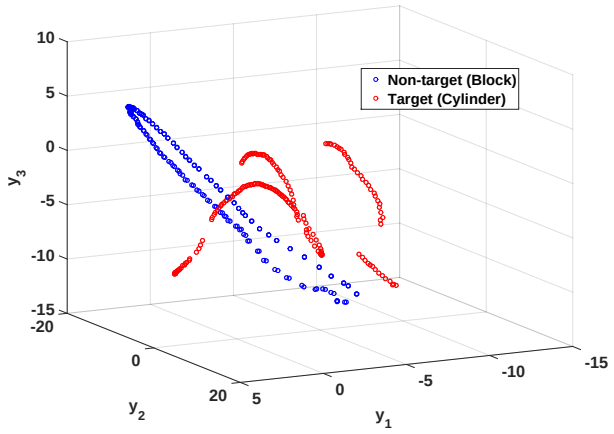

Figure 5.13: Distribution of estimated latent variables for SAS Images, MPPCA method with $L = 2$ and $d = 3$.

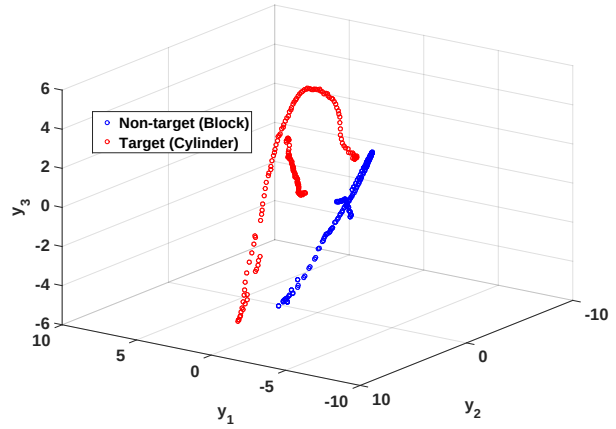

Figure 5.14: Distribution of estimated latent variables for SAS Images, MFM method with $L = 2$ and $d = 3$.

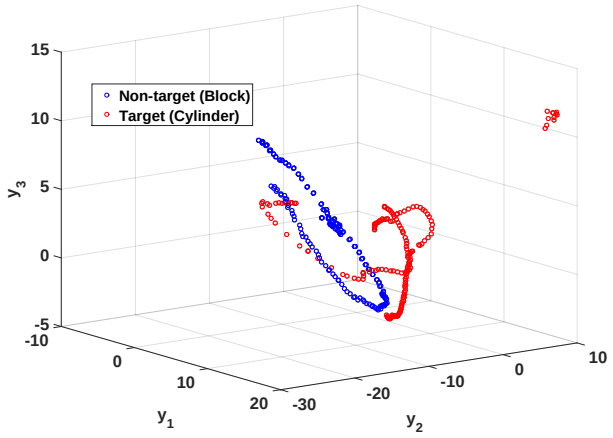

Figure 5.15: Distribution of estimated latent variables for SAS Images, MPPCA method with $L = 3$ and $d = 3$.

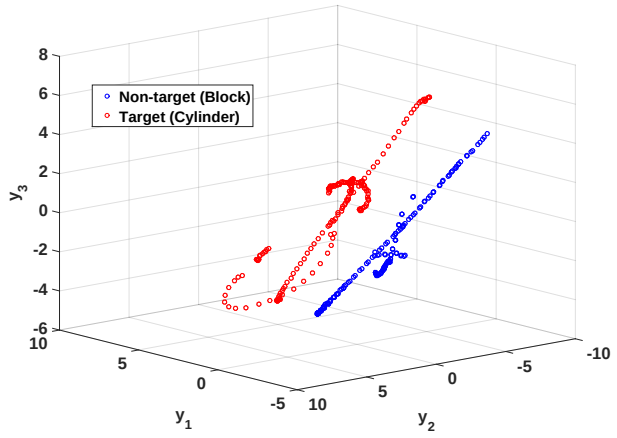

Figure 5.16: Distribution of estimated latent variables for SAS Images, MFM method with $L = 3$ and $d = 3$.

There are two main observations in the resultant distribution of low-dimensional features of MFM and PPCA/MPPCA. First, low-dimensional features of MFM method are linearly disjoint and thus suitable for classification whereas there is no such property for features of MPPCA. Second, as we increase the number of linear models, L, the more scattered the features become. Each cluster of features relates to one linear model.

As mentioned before, since we had finite number of test samples, sample average of risk of classification (which is greater than zero) is a better measure to quantify classification performance than empirical probability of correct classification, which is 100%. However, SVM with linear kernel also achieved the same probability of correct classification.

According to asymptotic behaviors discussed before, we expect to be able to perform classification based on the estimated latent variables since the expected clustering risk is negligible. It is fruitful to investigate this property by showing the linear discriminant function in $\mathbf{y}$ domain as well as estimated features. Figures 5.17 and 5.18 show the distribution of estimated latent variables using MFM method for $d = 3$ and $L = 2, 3$.

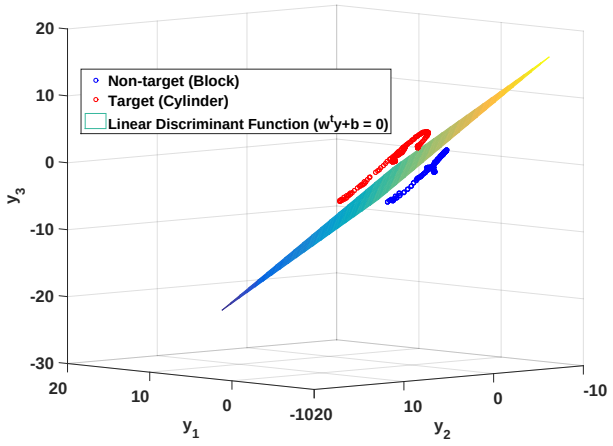

Figure 5.17: Distribution of estimated latent variables for MFM method and associated linear discriminant function, $L = 2$ and $d = 3$ case.

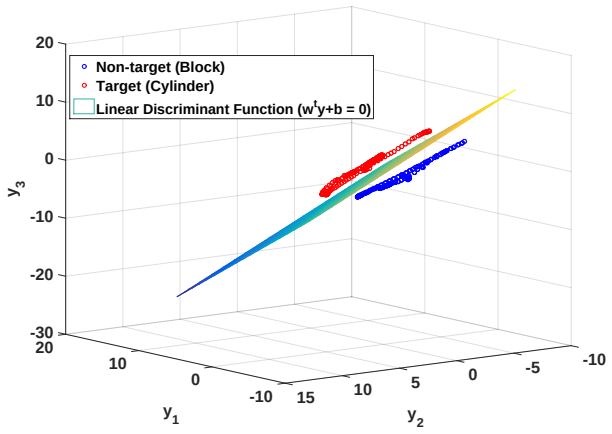

Figure 5.18: Distribution of estimated latent variables for MFM method and associated linear discriminant function, $L = 3$ and $d = 3$ case.

As it was expected, not only we are able to achieve the same empirical probability of correct classification, 100%, from both data samples and estimated latent variables but also the same empirical classification rates.

5.4 Results on Facial Image Database

The facial database consists of two sets of images. Each set contains of two series of 93 images of the same person at different poses. In the original database, there are 15 people with or without glasses and having various skin color. The pose, or head orientation is determined by 2 angles (h, v), which vary from -90 degrees to $+90$ degrees. Figures (5.19) and (5.20) give 15 different examples of two individuals.

Figure 5.19: Samples of Facial Images of Individual 1.

Figure 5.20: Samples of Facial Images of Individual 2.

All images have been taken using the FAME Platform of the PRIMA Team in INRIA Rhone-Alpes [38]. To obtain different poses, they put markers in the whole room. Each marker corresponds to a pose (h, v). The whole set of markers covers a half-sphere in front of the person. In order to obtain the face in the center of the image, the person is asked to adjust the chair to see the device in front of him. After this initialization phase, person is asked to stare successively at 93 markers, without moving his eyes.

Since MFM method provides binary classifier, we had to restrict our data set to two individuals only. Each image is of size 288×384 pixels which is subsequently sub-sampled to 48×64 images and then vetorized to yield data vectors of dimension $D = 3072$. The goal of classification is to identify each individual correctly. Two fold cross-validation method is used to evaluate the results. For this experiment, the dimension of the low-dimensional features is chosen to be $d = 6, 12, 18$ and number of linear model $L = 1, 2, 3$.

Table 5.4: Performance measures of the MFM method for different choices of d and L - facial image data set.

L	d	$\hat{R}_{dim.}$	$\hat{R}^{\mathbf{x}}_{clas.}$	$\hat{R}^{\mathbf{y}}_{clas.}$	$\hat{R}_{clus.}$	$\hat{P}^{\mathbf{x}}_{cc}$	$\hat{P}^{\mathbf{y}}_{cc}$	SER
1	6	0.0153	0.0362	0.0360	0	0.995	0.995	17.35
2	6	0.0120	0.0351	0.0350	1.0711e-06	0.995	0.995	18.38
3	6	0.0115	0.0365	0.0364	6.3916e-06	0.997	0.997	18.97
1	12	0.0151	0.0318	0.0318	0	0.997	0.997	18.95
2	12	0.0140	0.0315	0.0314	9.8115e-05	0.997	0.997	19.98
3	12	0.0117	0.0297	0.0297	4.3435e-10	0.997	0.997	20.83
1	18	0.0152	0.0303	0.0302	0	0.997	0.997	19.98
2	18	0.0133	0.0283	0.0282	6.1415e-18	0.997	0.997	21.20
3	18	0.0123	0.0285	0.0284	7.0485e-13	0.997	0.997	23.19

Table 5.4 provides the experimental results of applying MFM to facial image data set. Results are presented in terms of performance measures, e.g., sample average of dimensionality reduction, $\hat{R}_{dim.}$, and clustering, $\hat{R}_{clus.}$, risks, SER in dB, sample average of classification risk and probabilities of correct classification based on data samples $\mathbf{x}_n$, $\hat{R}^{\mathbf{x}}_{clas.}$ and $\hat{P}^{\mathbf{x}}_{cc}$, or based on estimated low-dimensional features $\hat{\mathbf{y}}_n$, $\hat{R}^{\mathbf{y}}_{clas.}$ and $\hat{P}^{\mathbf{y}}_{cc}$.

Table 5.5: Performance measures of the PPCA ($L = 1$) and MPPCA ($L > 1$) methods for different choices of d and L - facial image data set.

L	d	$\hat{R}_{dim.}$	$\hat{R}_{clus.}$	SER
1	6	0.0020	0	17.36
2	6	0.0214	5.5298e-04	18.28
3	6	0.0131	1.6792e-04	18.97
1	12	0.0027	0	18.95
2	12	0.0140	3.1350e-05	20.03
3	12	0.0124	5.6196e-09	20.84
1	18	0.0035	0	19.99
2	18	0.0169	1.7853e-04	21.28
3	18	0.0118	2.9108e-49	22.69

Table 5.5 provides the dimensionality reduction and clustering risks, reconstruction signal to noise ratio for PPCA and MPPCA algorithms on facial data set. For this experiment we can make mostly the same conclusions as in SAS data set. For instance, although both MFM and MPPCA employ soft clustering mechanism, due to small clustering risk they behave like a hard clustering method. However, one major observation in this experiment is that classification using estimated latent variables leads to smaller risk. The reason is that the optimality of classification based on data samples holds just for the *training data*. But the result shows that classification based on low-dimensional features has a better *generalization* property (compare $\hat{R}^x_{clas.}$ with $\hat{R}^y_{clas.}$).

In order to evaluate the reconstruction performance visually, we chose a fixed selection of images from both individuals and mapped them to d-dimensional latent space and then reconstructed them. Then, we studied the effects of d and model complexity L on the

reconstruction SER. For $L = 1$ case, there were no tangible difference between MFM and PPCA hence just MFM results are presented by Figures 5.22-5.24 (for $d = 6, 12, 18$). Figures 5.25-5.28 show the reconstructed facial images using both MFM and MPPCA methods for $d = 6$ and $L = 2, 3$.

Figure 5.21: An specific selection of individual 1 and 2 facial images.

Figure 5.22: Reconstructed facial images using FM method with $L = 1$ and $d = 6$.

Figure 5.23: Reconstructed facial images using FM method with $L = 1$ and $d = 12$.

Figure 5.24: Reconstructed facial images using FM method with $L = 1$ and $d = 18$.

Figure 5.25: Reconstructed facial images using MFM method with $L = 2$ and $d = 6$.

Figure 5.26: Reconstructed facial images using MFM method with $L = 3$ and $d = 6$.

Figure 5.27: Reconstructed facial images using MPPCA method with $L = 2$ and $d = 6$.

Figure 5.28: Reconstructed facial images using MPPCA method with $L = 3$ and $d = 6$.

Similar to the results in SAS dataset, the obvious conclusion from these experiments is that increasing the dimension of latent variables, d, leads to a better reconstruction quality in terms of SER (compare cases where $d \times L$ are equal in Tables 5.5 as well as 5.4). However, this quality is not visually tangible since facial features were completely blurred. Although increasing the number of linear models helps to achieve a better reconstruction too, increasing d seems to be more effective. Generally, MPPCA leads to a better reconstructed images

with higher reconstructed SER. The main counter example was $d = 18$ and $L = 3$ case where MFM lead to a much better SER than MPPCA since the parameters were initialized 5 different times (in parameters estimation) and we selected the one corresponds to the maximum likelihood of all. However, in both cases they have poor quality due to lack of enough training data since the inherent degrees of freedom in this data set is large (3 for face locations and 2 for head angle and etc.).

As mentioned before, the main distinction between MFM and MPPCA is the assumption of latent variable distribution for different classes with premise of getting generative-discriminative features. It is worthy to compare the distribution of estimated latent variables of MFM with PPCA/MPPCA for this experiment. Figures 5.29-5.34 provide estimated low dimensional features using both MPPCA/PPCA and MFM methods, for $d = 3$ and $L = 1, 2, 3$, respectively.

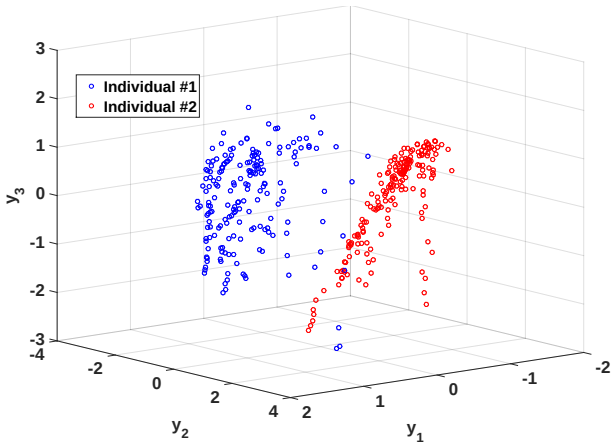

Figure 5.29: Distribution of estimated latent variables for facial data set, PPCA method with $d = 3$.

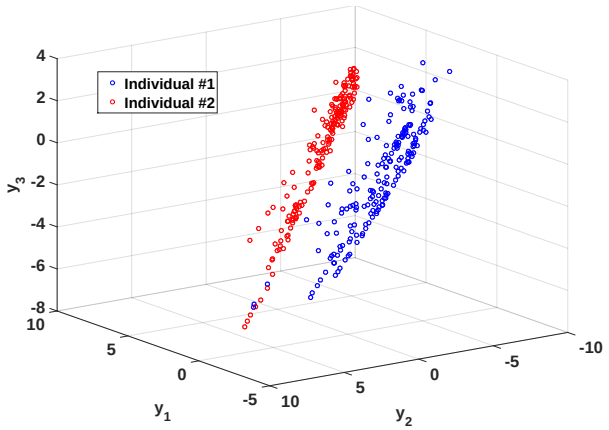

Figure 5.30: Distribution of estimated latent variables for facial data set, FM method with $L = 1$ and $d = 3$.

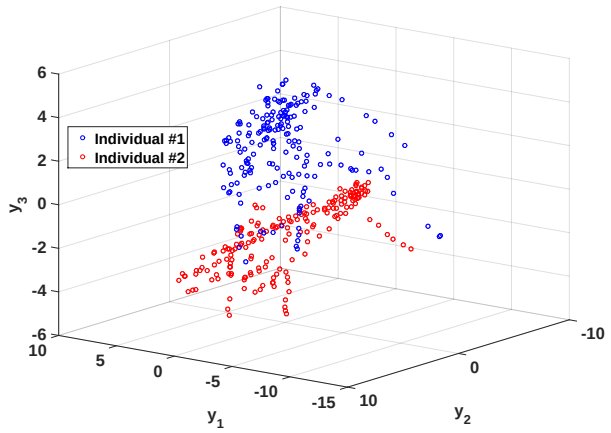

Figure 5.31: Distribution of estimated latent variables for facial data set, MPPCA method with $L = 2$ and $d = 3$.

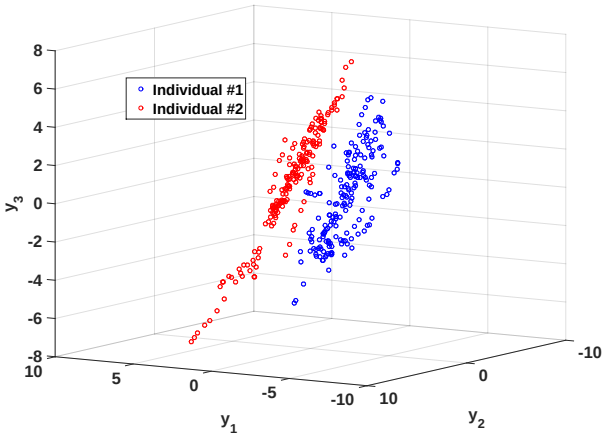

Figure 5.32: Distribution of estimated latent variables for facial data set, MFM method with $L = 2$ and $d = 3$.

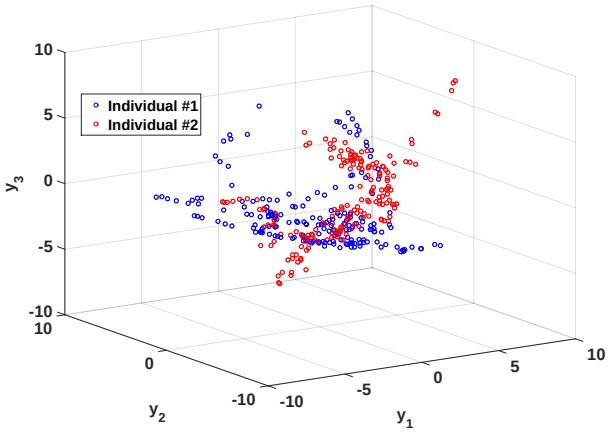

Figure 5.33: Distribution of estimated latent variables for facial data set, MPPCA method with $L = 3$ and $d = 3$.

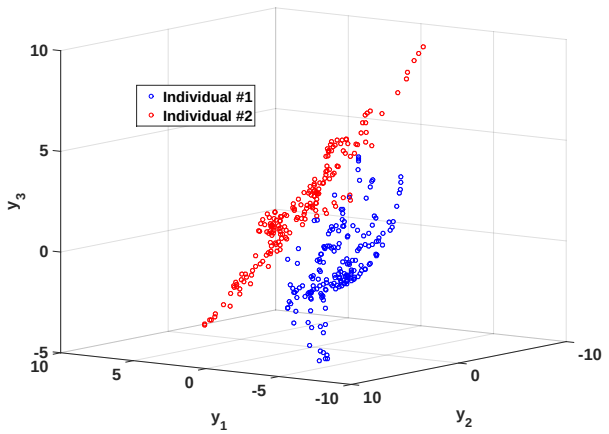

Figure 5.34: Distribution of estimated latent variables for facial data set, MFM method with $L = 3$ and $d = 3$.

In all the cases, the low-dimensional features of MFM method are linearly disjoint and thus suitable for classification whereas there is no such property for those of MPPCA. Moreover, classification risk is a more conservative measure than probability of correct classification as it yields to worse results (if we interpret it as probability of mis-classification). Since we had finite number of test samples, sample average of expected risk of classification (which is about

0.03) is a better measure to quantify classification performance than empirical probability of correct classification, which is 99%. However, SVM with linear kernel achieved slightly a better probability of correct classification of 100% due to its optimality in finding maximum margin discriminant hyperplane.

According to asymptotic behavior discussed before, we expect to be able to perform classification based on estimated latent variables (or low-dimensional features) since the expected clustering risk is negligible. It is fruitful to investigate this property by showing the linear discriminant function in $\mathbf{y}$ domain as well as estimated features. Figures 5.35 and 5.36 show the distribution of estimated latent variables for facial data set and associated linear discriminant function using MFM method with $d = 3$ and $L = 2, 3$.

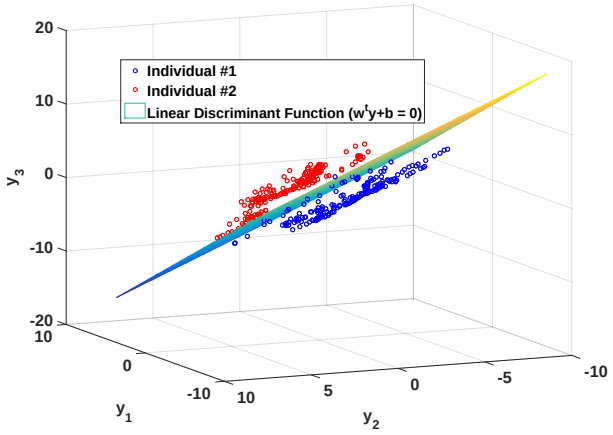

Figure 5.35: Distribution of estimated latent variables for facial data set and associated linear discriminant function, MFM method with $L = 2$ and $d = 3$.

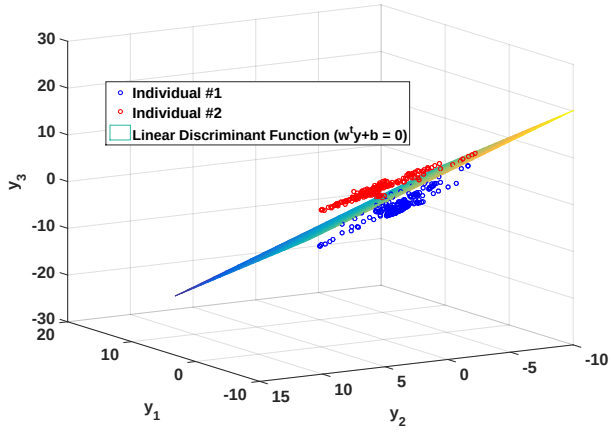

Figure 5.36: Distribution of estimated latent variables for facial data set associated linear discriminant function, MFM method with $L = 3$ and $d = 3$.

The figures illustrates an important property of resultant features of MFM. Not only the relative spatial coordinates of these features are useful for data visualization but also by comparing them with discriminant function we can classify them as well. Moreover, we are able to achieve the same empirical probability of correct classification, 99%, from both data samples and estimated latent variables.

5.5 Conclusion

In this chapter, we applied the proposed MFM method to two datasets, namely SAS and facial image data sets, to assess and compare its performance in clustering, dimensionality reduction, reconstruction and classification against PPCA/MPPCA and SVM with linear kernel, respectively.

Classification performance gives a promising results especially when the data is linearly separable. SVM with linear kernel is the optimal linear classifier as it provides the maximum margin hyperplane to classify the labeled data. Both of the dataset were linearly separable since we achieved perfect, 100%, probability of correct classification with SVM. However, MFM provides competitive results as well. One of the advantage of MFM over SVM is that it provides a classification risk which is a measure of classification confidence.

PPCA and MPPCA ultimate goals were data modeling and reconstruction. Comparing MFM dimensionality reduction and reconstruction with PPCA/MPPCA shows interesting resemblance. Occasionally MFM outperformed both of PPCA and MPPCA methods. For $L = 1$ case, the reason could be the inherent linear separability of the data which didn't affect reconstruction performance of MFM. For $L > 1$ case, the reason might be due to local optimality of both approaches which might prevent them to hit the best achievable parameters.

For both MFM and MPPCA methods, increasing dimension of feature space and model complexity, d or L, leads to a better generative features as they give better reconstructed images. Meanwhile, increasing model complexity didn't have detrimental effects on classification. But, in order to offer generalization ability, it needs more data samples to train on while avoid overfitting problem.

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

6.1 Conclusions

In general, dimensionality reduction can bring an improved understanding of the data whenever the intrinsic dimensionality of a data set is smaller than the actual one. Dimensionality reduction can also be seen as a feature extraction or mapping for representing the data in a different coordinate system. The fundamental assumption that justifies dimensionality reduction is that the data actually lies, at least approximately, on a low dimensional manifold of dimension $d \ll D$ that needs to be discovered. The goal of dimensionality reduction is to find a representation of that manifold which will allow us to obtain a low-dimensional, compact representation of the data. Considering joint classification and dimensionality reduction problem, one might be interested in extracting low-dimensional features that are truly representative of the properties of the original high-dimensional data and also determine its label at the same time. Finding a solution to relate dimensionality reduction with classification, motivated the present work to develop a supervised probabilistic approach for analyzing labeled high dimensional data with complex structures by exploiting a set of low dimensional latent variables.

The popular nonlinear dimensionality reduction and manifold learning methods, such as Isometric Feature Mapping (ISOMAP) [11], Maximum Variance Unfolding (MVU) or semi-definite embedding [12], Locally Linear Embedding (LLE) [13], and Laplacian Eigenmaps [14], suffer from some serious drawbacks that are:

1. The low-dimensional subspace provides either a generative representation of the original data, but not both discriminative and generative.

2. There usually is no explicit mapping for out-of-sample data.

Comparing our method to the probabilistic methods, such as Probabilistic PCA [2], Mixture of Probabilistic PCA [1] and Generative Topographic Mapping [17]. the same drawback in item 1 above holds for these systems.

Chapter 2 reviewed latent variable models where both the latent and the observed variables are continuous, going further than the famous linear model of factor analysis [24] [16]. We began with the most common example of a latent variable model, i.e. the statistical Factor Analysis. Different shortcomings of this method, such as having multiple local likelihood maxima and lack of additivity property were discussed. With a simple change in Factor Analysis model, PPCA was then derived from the perspective of density estimation. The association of a probability model with PCA offers the tempting prospects. First, being able to compare with other density estimation techniques and facilitate statistical testing through the resultant likelihood. Second, Bayesian inference methods could be applied by combining the likelihood with a prior. Third, the value of the probability density function could be used as a measure of the "degree of novelty" of a new data point. And finally, being able to model complex data structures with a combination of local PCA models through the mechanism of a mixture of probabilistic principal component analyzers. The probabilistic model for PCA was exploited to combine local PCA models within the framework of a probabilistic mixture in which all the parameters are determined from ML using an EM algorithm [1]. A possible disadvantage of the probabilistic approach to combining local PCA models is that, by optimizing a likelihood function, the MPPCA does not directly minimize squared reconstruction error. For applications where this is the key requirement, algorithms which explicitly minimize reconstruction error should be expected to perform better.

In Chapter 3, we discussed the general framework of the proposed model which provides a supervised probabilistic approach for analyzing labeled high dimensional data by exploiting a set of low dimensional latent variables. Mixture of factor models relate latent variable to observable variable, i.e. mixture of linear models is defined to map latent variable to ambient

data space. A linear classifier is then built in the latent domain to perform two-class decision-making. The pdf of the low dimensional latent variable associated with each class was shown to be a squashed multivariate normal leading to linear separability of the estimated low-dimensional latent variables (features). Using the transformation from latent space to data space leads to data synthesis or reconstruction whereas estimating the latent variable amounts to dimensionality reduction and data clustering. It was shown that resultant low dimensional feature is a collaborative estimates of each linear model weighted by likelihood of the model. In the special case of small clustering risk or when $L = 1$, a simplified least squares (LS) version of reconstruction is provided to reconstruct data samples. Finally, we develop a Bayesian classification algorithm, based on data samples as well as estimated low dimensional features. Same as dimensionality reduction, the estimated class label based on data sample is also a collaborative estimates of each linear model weighted by likelihood of the model. Due to its optimality, which leads to the minimum classification risk, it is preferable to estimate class labels based on data samples. However, for small clustering or small dimensionality reduction risk, one can classify data samples based on their estimated latent variables as well.

Chapter 4 presented a mixture of EM and steepest descent algorithms to derive the ML estimates of the unknown parameters of the model. It was shown that parameters associated with dimensionality reduction can be estimated using the EM algorithm whereas those of the classifier can be estimated using a steepest descent algorithm. Squashed multivariate normal distribution and its moments, using linear boundary, helped us finding analytical solutions in the E-Step of EM algorithm for the dimensionality reduction parameters. It encompasses general cases of multivariate truncated normal and skew-normal distributions. Classification parameters appeared nonlinearly in the formulation and thus we were unable to estimate them using EM. Therefore, steepest descent algorithm was adopted to estimate these parameters.

In Chapter 5, we applied the developed *Mixture of Factor Models* (MFM) to two datasets, synthetic aperture sonar (SAS) images and facial images. The first data set consists of 181 SAS images of model-generated underwater objects. The second data set consists of 186 facial images of different individuals. Classification performance is compared to that of SVM with linear kernel whereas the dimensionality reduction and signal reconstruction are compared to those of PPCA/MPPCA. Classification performance gives promising results especially when the data is linearly separable. SVM with linear kernel is the best linear classifier as it provides the maximum margin hyperplane to classify data. The probability of correct classification for SVM and our proposed MFM methods was found to be 100% on both datasets. Sample classification risk, which can be interpreted as mis-classification probability, is a more conservative measure provided by MFM. PPCA and MPPCA ultimate goal were data modeling and reconstruction. Comparing MFM dimensionality reduction and reconstruction performance with those of PPCA/MPPCA shows interesting resemblance. Occasionally MFM outperformed both of PPCA and MPPCA methods (in SER sense). For $L = 1$ case, the reason could be the inherent linear separability of the data which didn't affect reconstruction performance of MFM. For $L > 1$ case, the reason might be due to local optimality of both approaches which might prevent them to hit the best achievable parameters (compare resultant SER for $d = 18$ and $L = 3$ case from applying MFM on facial dataset with that of MPPCA). For both MFM and MPPCA methods, increasing model complexity, d or L, led to a better generative features as they gave better reconstructed images. It seemed that inherent dimensionality of SAS images were small ($d = 3$) due to good quality of reconstruction visual appearance whereas for the facial images this is not true as the facial features were completely lost in the reconstruction. The low dimensional features extracted by MFM have discriminatory property which can not be shared by PPCA/MPPCA. Meanwhile, increasing model complexity didn't have detrimental effects on the classification performance. But, in order to have good generalization, it needs more data samples to train while avoiding overfitting problem.

6.2 Future Work

Although the proposed probabilistic method based on mixture of factor models offers a promising solution to classification and dimensionality reduction problems, there is still room for improvements in many different aspects which can be pursued in the future. These include but are not limited to:

- MFM method assumes a linear discriminant function which may affect the quality of classification if data in the feature space is not linearly separable. Generalizing it to a nonlinear function in the latent space can offer more promising results.

- Both EM and steepest descent algorithms are batch and iterative in nature. Another item for future research is to develop online optimization method for streaming data.

- Assuming priors for the parameters and employing Bayesian inference can be helpful to avoid overfitting problem, since Bayesian inference takes average of the objective function (for instance likelihood function) w.r.t the posterior distribution of the parameters. This approach can be an answer to the situations where training and testing data come from different mediums, e.g., training on model-generated SAS images and testing on the real SAS data.

- For $L > 1$ models, estimated parameters are different in each experiment due to existence of multiple local maxima in the likelihood function. Develop a method for merging these estimated parameters so that the new set of parameters yield a better likelihood.

- Introduce a control parameter to stress either "generative" or "discriminative" properties of the low dimensional features. For instance in the $L = 1$ case, we showed that the likelihood function can be written as the product of two terms L_G and L_D. By elaborating the significance of these components this goal can be achieved, i.e. $Likelihood \propto (L_D)^{\alpha} \times L_G$.

- The proposed method can be generalized to suit multi-class problems, while using a linear discriminant function in latent space.
- We can employ different methods, such as [39] and [40], to improve the probability of hitting the global maximum of the likelihood function.

REFERENCES

[1] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural computation*, vol. 11, no. 2, pp. 443–482, 1999.

[2] ——, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[3] R. E. Bellman, "Adaptive control processes," *Princeton, NJ*, 1961.

[4] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[5] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," pp. 163–222, 2012.

[6] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[7] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[9] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[10] L. K. Saul, K. Q. Weinberger, J. H. Ham, F. Sha, and D. D. Lee, "Spectral methods for dimensionality reduction," *Semisupervised learning*, pp. 293–308, 2006.

[11] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science Journal*, vol. 290, pp. 2319–2323, 2000.

[12] K. Q. Weinberger and L. K. Saul, "An introduction to nonlinear dimensionality reduction by maximum variance unfolding," vol. 6, pp. 1683–1686, 2006.

[13] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science Journal*, vol. 290, pp. 2323–2326, 2000.

[14] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[15] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC press, 2000.

[16] D. J. Bartholomew, M. Knott, and I. Moustaki, *Latent variable models and factor analysis: A unified approach*. John Wiley & Sons, 2011.

[17] C. M. Bishop, M. Svensén, and C. K. Williams, "GTM: The generative topographic mapping," *Neural computation*, vol. 10, pp. 215–234, 1998.

[18] C. M. Bishop, "Bayesian PCA," *Advances in neural information processing systems*, pp. 382–388, 1999.

[19] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning Journal*, vol. 20, no. 3, pp. 273–297, 1995.

[21] C. M. Bishop, "Latent variable models," pp. 371–403, 1998.

[22] ——, "Pattern recognition," *Machine Learning Journal*, 2006.

[23] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[24] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, pp. 502–516, 1989.

[25] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley & Sons, 2004, vol. 46.

[26] H. Attias, "Independent factor analysis," *Neural computation*, vol. 11, pp. 803–851, 1999.

[27] A. T. Basilevsky, *Statistical factor analysis and related methods: theory and applications*. John Wiley & Sons, 2009, vol. 418.

[28] T. W. Anderson, "Asymptotic theory for principal component analysis," *The Annals of Mathematical Statistics*, vol. 34, no. 1, pp. 122–148, 1963.

[29] L. L. Scharf and D. W. Tufts, "Rank reduction for modeling stationary signals," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 35, no. 3, pp. 350–355, 1987.

[30] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[31] R. Tibshirani, "Principal curves revisited," *Statistics and Computing*, vol. 2, pp. 183–190, 1992.

[32] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *American Institute of Chemical Engineers (AIChE) journal*, vol. 37, pp. 233–243, 1991.

[33] Z. Ghahramani, G. E. Hinton *et al.*, "The EM algorithm for mixtures of factor analyzers," Technical Report CRG-TR-96-1, University of Toronto, Tech. Rep., 1996.

[34] W. C. Horrace, "Some results on the multivariate truncated normal distribution," *Journal of Multivariate Analysis*, vol. 94, pp. 209–221, 2005.

[35] A. Azzalini and A. Dalla Valle, "The multivariate skew-normal distribution," *Biometrika*, vol. 83, no. 4, pp. 715–726, 1996.

[36] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, 2003.

[37] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," pp. 355–368, 1998.

[38] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial structures," pp. 1–9, 2004.

[39] D. Karlis and E. Xekalaki, "Choosing initial values for the em algorithm for finite mixtures," *Computational Statistics & Data Analysis*, vol. 41, no. 3, pp. 577–590, 2003.

[40] A. O. Griewank, "Generalized descent for global optimization," *Journal of optimization theory and applications*, vol. 34, pp. 11–39, 1981.

[41] G. Rodriguez-Yam, R. A. Davis, and L. L. Scharf, "Efficient gibbs sampling of truncated multivariate normal with application to constrained linear regression," *Unpublished manuscript*, 2004.

[42] J. A. Gubner, *Probability and random processes for electrical and computer engineers*. Cambridge University Press, 2006.

[43] C. K. Williams and C. E. Rasmussen, "Gaussian processes for machine learning," *the MIT Press*, vol. 2, no. 3, p. 4, 2006.

APPENDIX A

SQUASHED MULTIVARIATE NORMAL DISTRIBUTION

A.1 Definitions and Preliminaries

A squashed multivariate normal distribution is the probability distribution of a multivariate normally distributed random vector whose probability density function is skewed by a squashing function. After presenting the formal definition of this random variable and its statistics in special cases, we conclude with the proof of each part.

Let $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{R})$ and an auxiliary binary random variable z with $f(z = 1|\mathbf{y}) = \psi(g(\mathbf{y}))$, where $\psi(.)$ is an arbitrary squashing function and $g(\mathbf{y}) = 0$ specifies squashing boundaries. Then $\mathbf{y}$ conditioned on $z = 1$ (as well as $z = 0$) has a squashed multivariate normal distribution of form

$$f(\mathbf{y}|z = 1) = \frac{\psi(g(\mathbf{y}))}{E[\psi(g(\mathbf{y}))]} \frac{1}{\sqrt{(2\pi)^d |\mathbf{R}|}} \exp\Big(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \mathbf{R}^{-1} (\mathbf{y} - \boldsymbol{\mu}) \Big)$$

where

$$E[\psi(g(\mathbf{y}))] = \int \psi(g(\mathbf{y})) f(\mathbf{y}) d\mathbf{y}. \tag{A-1}$$

First and second order moments of squashed multivariate normal are closely related to the original multivariate normal via $\log E[\psi(g(\mathbf{y}))]$ as follows (see Section A.2),

$$E[\mathbf{y}|z = 1] = \boldsymbol{\mu} + \tilde{\boldsymbol{\mu}}, \tag{A-2}$$

$$E[\mathbf{y}\mathbf{y}^T|z = 1] = \mathbf{R} + \boldsymbol{\mu}\boldsymbol{\mu}^T + \tilde{\boldsymbol{\mu}}\boldsymbol{\mu}^T + \boldsymbol{\mu}\tilde{\boldsymbol{\mu}}^T + \tilde{\mathbf{R}} \tag{A-3}$$

where $\tilde{\boldsymbol{\mu}} = \mathbf{R}\nabla_{\boldsymbol{\mu}} \log E[\psi(g(\mathbf{y}))]$, $\tilde{\mathbf{R}} = 2\mathbf{R}\nabla_{\mathbf{R}} \log E[\psi(g(\mathbf{y}))]\mathbf{R}$, and $\nabla_{\boldsymbol{\mu}}$ and $\nabla_{\mathbf{R}}$ are gradients w.r.t $\boldsymbol{\mu}$ and $\mathbf{R}$ respectively. (See the proofs in Section A.2)

Special Cases

1. **Truncated Multivariate Normal:** When using Heaviside squashing function, $\psi(t) = \mathbf{1}_{(0,+\infty)}(t)$, this distribution boils down to the general truncated multivariate normal distribution, [41], and we have

$$\begin{aligned} E[\psi(g(\mathbf{y}))] &= E[\mathbf{1}_{(0,+\infty)}(g(\mathbf{y}))] \\ &= \Pr(g(\mathbf{y}) > 0). \end{aligned}$$

   which is the volume under one side of the multivariate normal distribution.

2. **Linearly Truncated Multivariate Normal:** If we use $\psi(g(\mathbf{y})) = \mathbf{1}_{(0,\infty)}(\mathbf{w}^T\mathbf{y} + b)$, then we have

$$E[\psi(g(\mathbf{y}))] = \Phi(\frac{\mathbf{w}^T\boldsymbol{\mu} + b}{\sqrt{\mathbf{w}^T\mathbf{R}\mathbf{w}}}) \tag{A-4}$$

   where $\Phi(\cdot)$ is the CDF of a standard normal distribution. This expression can easily be proven by using the fact that $\mathbf{w}^t\mathbf{y} + b \sim \mathcal{N}(\mathbf{w}^t\boldsymbol{\mu} + b, \mathbf{w}^t\mathbf{R}\mathbf{w})$. First and second order moments can then be derived by calculating the gradient of (A-4) w.r.t $\boldsymbol{\mu}$ and $\mathbf{R}$ and replace them in equations (A-2) and (A-3). These gradients are,

$$\nabla_{\boldsymbol{\mu}} \log E[\psi(g(\mathbf{y}))] = \frac{1}{\Phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{\mathbf{w}^T\mathbf{R}\mathbf{w}}})} \frac{\phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{\mathbf{w}^T\mathbf{R}\mathbf{w}}})}{\sqrt{\mathbf{w}^T\mathbf{R}\mathbf{w}}}\mathbf{w}, \tag{A-5}$$

$$\nabla_{\mathbf{R}} \log E[\psi(g(\mathbf{y}))] = -\frac{\mathbf{w}^T\boldsymbol{\mu} + b}{2\Phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{\mathbf{w}^T\mathbf{R}\mathbf{w}}})} \frac{\phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{\mathbf{w}^T\mathbf{R}\mathbf{w}}})}{(\mathbf{w}^T\mathbf{R}\mathbf{w})^{\frac{3}{2}}})\mathbf{w}\mathbf{w}^T, \tag{A-6}$$

   where $\phi(.)$ is the pdf of a standard normal distribution.

3. **Linearly Φ-Squashed Multivariate Normal:** In case of using $\psi(t) = \Phi(t)$ and $g(\mathbf{y}) = \mathbf{w}^T\mathbf{y} + b$, this distribution is a special case called Linearly Φ-Squashed Multivariate Normal Distribution, which is closely related to skew-normal distribution [35], and we have,

$$\begin{aligned} E[\psi(g(\mathbf{y}))] &= E[\Phi(\mathbf{w}^T\mathbf{y} + b)] \\ &= \Phi(\frac{\mathbf{w}^T\boldsymbol{\mu} + b}{\sqrt{1 + \mathbf{w}^T\mathbf{R}\mathbf{w}}}) \end{aligned} \tag{A-7}$$

where the proof is presented in Section A.2. The main difference between linearly Φ-squashed normal and its truncated counterpart is that it doesn't have any singularity problems due to the positive denominator of $1 + \mathbf{w}^T\mathbf{R}\mathbf{w}$.

First and second order moments can be derived by calculating the gradient of equation (A-7) w.r.t $\boldsymbol{\mu}$ and $\mathbf{R}$, i.e.

$$\nabla_{\boldsymbol{\mu}} \log E[\psi(g(\mathbf{y}))] = \frac{1}{\Phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}\mathbf{w}}})} \frac{\phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}\mathbf{w}}})}{\sqrt{1 + \mathbf{w}^T\mathbf{R}\mathbf{w}}}\mathbf{w} \tag{A-8}$$

$$\nabla_{\mathbf{R}} \log E[\psi(g(\mathbf{y}))] = -\frac{\mathbf{w}^T\boldsymbol{\mu} + b}{2\Phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}\mathbf{w}}})} \frac{\phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}\mathbf{w}}})}{(1 + \mathbf{w}^T\mathbf{R}\mathbf{w})^{\frac{3}{2}}})\mathbf{w}\mathbf{w}^T, \tag{A-9}$$

and replace them in equations (A-2) and (A-3). Similarly, first and second order moments of $\mathbf{y}$ given $z = 0$ can be derived by substituting $\Phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}\mathbf{w}}})$ with $\Phi(\frac{\mathbf{w}^T\boldsymbol{\mu}+b}{\sqrt{1+\mathbf{w}^T\mathbf{R}\mathbf{w}}}) - 1$ in the resultant expressions for those of $\mathbf{y}$ given $z = 1$ case. Combining these two results yields the moments given by (3.4) and (3.5).

A.2 Proofs of Proposition 3.1

(a) We start by using Bayes' rule for $f(\mathbf{y}|z=1)$,

$$f(\mathbf{y}|z=1) = \frac{\psi(g(\mathbf{y}))}{\int \psi(g(\mathbf{y}))f(\mathbf{y})d\mathbf{y}} f(\mathbf{y})$$

where $\psi(g(\mathbf{y})) = f(z=1|\mathbf{y})$. Conditional distribution of $f(\mathbf{y}|z=0)$ is also squashed multivariate normal distribution,

$$f(\mathbf{y}|z=0) = \frac{1-\psi(g(\mathbf{y}))}{1-E[\psi(g(\mathbf{y}))]} f(\mathbf{y}).$$

Combining these two pdfs yields the conditional distribution of $f(\mathbf{y}|z)$ in (3.3).

(b) To find a closed form expression for the conditional mean of $\mathbf{y}$ given $z=1$,

$$E[\mathbf{y}|z=1] = \frac{1}{E[\psi(g(\mathbf{y}))]} \int \mathbf{y}\psi(g(\mathbf{y}))f(\mathbf{y})d\mathbf{y} \tag{A-10}$$

where $E[\psi(g(\mathbf{y}))] = \int \psi(g(\mathbf{y}))f(\mathbf{y})d\mathbf{y}$. Now, consider taking the gradient of $E[\psi(g(\mathbf{y}))]$ w.r.t $\boldsymbol{\mu}$. Note that $\nabla_{\boldsymbol{\mu}} f(\mathbf{y}) = \mathbf{R}^{-1}(\mathbf{y}-\boldsymbol{\mu})f(\mathbf{y})$. After simplifying terms, we have

$$\begin{aligned}\nabla_{\boldsymbol{\mu}} E[\psi(g(\mathbf{y}))] &= \int \mathbf{R}^{-1}(\mathbf{y}-\boldsymbol{\mu})\psi(g(\mathbf{y}))f(\mathbf{y})d\mathbf{y} \\ &= \mathbf{R}^{-1}\int \mathbf{y}\psi(g(\mathbf{y}))f(\mathbf{y})d\mathbf{y} - \mathbf{R}^{-1}\boldsymbol{\mu}E[\psi(g(\mathbf{y}))] \end{aligned} \tag{A-11}$$

Conditional mean of $E[\mathbf{y}|z=1]$ in (A-10) can be found by pre-multiplying by $\mathbf{R}$ and rearranging terms in (A-11).

To find a closed form expression for the conditional second order moment, $E[\mathbf{y}\mathbf{y}^t|z=1]$, we can write

$$E[\mathbf{y}\mathbf{y}^T|z=1] = \frac{1}{E[\psi(g(\mathbf{y}))]} \int \mathbf{y}\mathbf{y}^T\psi(g(\mathbf{y}))f(\mathbf{y})d\mathbf{y}. \tag{A-12}$$

Using the same approach as in (A-11) and taking gradient of $E[\psi(g(\mathbf{y}))]$ w.r.t $\mathbf{R}$, we get

$$\nabla_{\mathbf{R}} E[\psi(g(\mathbf{y}))] = -\frac{1}{2}\int \left(\mathbf{R}^{-1} - \mathbf{R}^{-1}(\mathbf{y}-\boldsymbol{\mu})(\mathbf{y}-\boldsymbol{\mu})^T\mathbf{R}^{-1}\right)\psi(g(\mathbf{y}))f(\mathbf{y})d\mathbf{y}. \tag{A-13}$$

Now, after pre- and post-multiplying (A-13) by $\mathbf{R}$, rearranging it and using (A-11), we can easily find the conditional second order moment $E[\mathbf{y}\mathbf{y}^t|z=1]$ as in (A-3).

Note:

For the linearly Φ-squashed normal distribution, the Law of the Unconscious Statistician (LOTUS) [42] can be used to simplify $E[\Phi(\mathbf{w}^T\mathbf{y}+b)]$ as follows,

$$\begin{aligned} E[\Phi(\mathbf{w}^T\mathbf{y}+b)] &= \int \Phi(\mathbf{w}^T\mathbf{y}+b)f(\mathbf{y})d\mathbf{y} \\ &= \int \Phi(x)\frac{1}{\sqrt{2\pi\mathbf{w}^T\mathbf{R}\mathbf{w}}}\exp\Big(-\frac{(x-\mathbf{w}^T\boldsymbol{\mu}-b)^2}{2\mathbf{w}^T\mathbf{R}\mathbf{w}}\Big)dx. \end{aligned}$$

It is fruitful to find a closed form expression for the general case of this integral, [43], i.e.

$$\int \Phi(\frac{x-m}{\sqrt{u^2}})\frac{1}{\sqrt{2\pi v^2}}\exp(-\frac{(x-n)^2}{2v^2})dx. \tag{A-14}$$

Substituting the definition of $\Phi(x)$ in the integral and simplifying terms, we have

$$\frac{1}{2\pi uv}\int_{-\infty}^{+\infty}\int_{-\infty}^{x}\exp\Big(-\frac{(y-m)^2}{2u^2}-\frac{(x-n)^2}{2v^2}\Big)dydx$$

With a simple variable substitution,

$$\begin{bmatrix} z \\ w \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} y \\ x \end{bmatrix} + \begin{bmatrix} n-m \\ -n \end{bmatrix}$$

it becomes the integral of a multivariate normal distribution

$$\begin{aligned} \int_{-\infty}^{n-m}\int_{-\infty}^{+\infty}\mathcal{N}(\boldsymbol{\mu}_{z,w},\mathbf{R}_{z,w})dwdz &= \int_{-\infty}^{n-m}\mathcal{N}(\mu_z,R_z)dz \\ &= \Pr(Z<n-m) \end{aligned} \tag{A-15}$$

where the composite mean, $\boldsymbol{\mu}_{z,w}$, and covariance, $\mathbf{R}_{z,w}$, are

$$\boldsymbol{\mu}_{z,w} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{R}_{z,w} = \begin{bmatrix} v^2 + u^2 & -v^2 \\ -v^2 & v^2 \end{bmatrix}.$$

The distribution of Z is normal with mean $\mu_z = 0$ and variance $R_z = v^2 + u^2$. Thus, (A-14) could be simplified by using (A-15),

$$\int \Phi\Big(\frac{x-m}{\sqrt{u^2}}\Big) \frac{1}{\sqrt{2\pi v^2}} \exp\Big(-\frac{(x-n)^2}{2v^2}\Big) dx = \Phi\Big(\frac{n-m}{\sqrt{v^2+u^2}}\Big).$$

By comparison ($u = 1$, $n = \mathbf{w}^T\mathbf{y} + b$, $m = 0$ and $v^2 = \mathbf{w}^T\mathbf{R}\mathbf{w}$), hence we can find a closed form expression for $E[\Phi(\mathbf{w}^T\mathbf{y} + b)]$ as,

$$E[\Phi(\mathbf{w}^T\mathbf{y} + b)] = \Phi\Big(\frac{\mathbf{w}^T\boldsymbol{\mu} + b}{\sqrt{1 + \mathbf{w}^T\mathbf{R}\mathbf{w}}}\Big).$$