Mixture of Bilateral-Projection Two-dimensional Probabilistic Principal Component Analysis

Fujiao Ju[1], Yanfeng Sun[1], Junbin Gao[2], Simeng Liu[1], Yongli Hu[1], Baocai Yin[3]

[1]College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China

[2]Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, NSW 2006, Australia

[3]Faculty of Electronic Information and Electrical Engineering, College of Computer Science,

Dalian University of Technology, Dalian 116024, China

{jufujiao2013, smliu}@emails.bjut.edu.cn; {yfsun,huyongli,ybc}@bjut.edu.cn; junbin.gao@sydney.edu.au

Abstract

The probabilistic principal component analysis (PPCA) is built upon a global linear mapping, with which it is insufficient to model complex data variation. This paper proposes a mixture of bilateral-projection probabilistic principal component analysis model (mixB2DPPCA) on 2D data. With multi-components in the mixture, this model can be seen as a 'soft' cluster algorithm and has capability of modeling data with complex structures. A Bayesian inference scheme has been proposed based on the variational EM (Expectation-Maximization) approach for learning model parameters. Experiments on some publicly available databases show that the performance of mixB2DPPCA has been largely improved, resulting in more accurate reconstruction errors and recognition rates than the existing PCA-based algorithms.

1. Introduction

Principle Component Analysis (PCA) [3] is one of popular dimensionality reduction methods widely used in image analysis [9, 11], pattern recognition [8, 14] and machine learning [13] for data analysis. It can be derived under algebraic framework. However, algebraic models don't have flexibility of providing confidence information of the model when dealing with noisy data. This is due to the absence of an associated probability density or generative model in algebraic framework.

To compensate the algebraic PCA drawbacks, Tipping and Bishop [19] firstly proposed a probabilistic PCA model, called PPCA. Under the probabilistic framework, PPCA takes advantage of Bayesian learning and inference by combining the likelihood with appropriate priors. As a result, the observed data are regarded as random variables, generated from a set of latent random variables which follow the Gaussian distribution of zero mean and identity covariance, with additive noises following a Gaussian distribution with zero mean and an isotropic covariance. Under such a probabilistic learning framework, the model parameters in PPCA can be easily solved by the maximum likelihood estimation (MLE). Much progress has been made based on PCA and PPCA in the last couple of decades [2, 5].

PPCA and standard PCA methods can be interpreted in many ways, one of which assumes that the observed high-dimensional data are generated from their low-dimensional factors through a linear model with the corruption of Gaussian noise. So those algorithms essentially use a linear model for representing the entire data in a low dimensional subspace. It may be insufficient to model data with large variation caused by, for example, pose, expression and lighting in face recognition. Thus the application scope of PPCA and PCA-based methods is necessarily somewhat limited by its global linearity assumption. An alternative improving paradigm is to model the complex manifold with a mixture of local linear PPCA sub-models. Thus the single PCA model could be extended to a mixture of such sub-models.

A number of 'mixture of PPCA' have been proposed in literature. The first work was done by Ghahramani and Hinton [7]. They presented an exact Expectation-Maximization (EM) algorithm for fitting the parameters of the mixture of factor analyzers. By constraining the error covariance to be a diagonal matrix whose elements are usually equal, the mixture of factor analyzers became the mixture of PPCA [20]. Bishop and Tipping [4] extended the mixture of PPCA model to achieve a hierarchical mixture model. Su and Dy [17] introduced an automated hierarchical mixture of PPCA algorithm, which utilizes the integrated classification likelihood as a criterion for splitting and stopping the addition of hierarchical levels. Kim *et al.* [12] proposed a fast and sub-optimal selection method of model order such as the number of mixture components and the number of PCA

bases for the PCA mixture model, consisting of a combination of many PCAs. In addition, under the assumption of the Student-t distribution, the related research includes the mixture model of Student-t components [15], which actually is a generalized mixture of Gaussian model without considering subspace structures, and more recent work such as the robust subspace mixture model [16], in which both the likelihood and the latent variables were supposed to follow the Student-t distribution and the EM algorithm was applied to the model. In 2005, Archambeau [1] discussed the robust models in the context of finite mixture models, and a similar work for the mixture of the robust Laplacians was presented in [6]. These mixture models are important as it enables one to model nonlinear relationships by aligning a collection of such local models.

The aforementioned models are concerned with vectorial data. In order to apply these methods to 2D data, a typical workaround way is to vectorize 2D data. Vectorizing 2D data not only results in very high-dimensional data, causing the problem of the curse of dimensionality [23], but also ignores valuable information on the spatial relationship among 2D data. Instead of using vectorization, PCA approaches for two-dimensional data (2DPCA) have been proposed [22, 24, 26], to generally extract features of 2D data under the assumption of Gaussian noises. Ju *et al*. [10] proposed a probabilistic 2DPCA model to deal with outlier noises by using Laplacian distribution. This model benefits outlier detection. Wang *et al*. [21] extended the probabilistic 2DPCA to a mixture of local probabilistic 2DPCA models (MP2DPCA). MP2DPCA offers a tempting prospect of being able to model data with complex variation.

MP2DPCA model regards each row vector of the 2D data as an observed sample and uses all rows to train the mixture model, resulting in mean vectors from the mixture model. This is essentially a unilateral projection based scheme, where only one side multiplication is taken into account. The unilateral scheme usually preserves the correlation information among the row/column vectors of the images and more parameters are needed to well represent an image. To tackle these problems, a bilateral-projection scheme is favored. In this study, our intention is to propose a mixture of bilateral-projection-based probabilistic 2DPCA (mixB2DPPCA) model. Different from MP2DPCA, we regard each 2D images as observed samples in their natural shape and reduce 2D dimensionality directly. The mixB2DPPCA has two major advantages: 1) The model makes use of structured information of 2D data and can be easily extended for high order tensorial data. All the algorithm derivations remain without major difficulties. 2) mix2DPPCA carries over all the advantages of the mixture of PPCA.

The remainder of the paper is organized as follows. In Section 2, the mixture of bilateral-projection two-dimensional probabilistic PCA model is introduced. The variational approximation approach for solving the model is presented in Section 3. In Section 4, some experimental results are conducted to evaluate the performance of the proposed model. Finally, conclusions are summarized in Section 5.

2. Mixture of Bilateral-Projection 2DPPCA Model (mixB2DPPCA)

In this section, we introduce the mixture of bilateral-projection probabilistic 2DPCA model. For the purpose, we introduce several notations. Let $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N\}$ be N independent and identical random samples with values in $\mathbb{R}^{p \times q}$. For $n = 1, ..., N$, we suppose that sample $\mathbf{X}_n$ is generated independently from a mixture of K underlying components with unknown probabilities $\pi_1, \pi_2, ..., \pi_K$,

$$p(\mathbf{X}_n | \mathbf{B}_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{X}_n | \mathbf{L}_k \mathbf{B}_n^{(k)} \mathbf{R}_k^T + \mathbf{M}_k, \sigma_k \mathbf{I}, \sigma_k \mathbf{I})$$

(1)

where $\mathbf{M}_k \in \mathbb{R}^{p \times q}$ is the mean matrix, π_ks satisfy $\pi_k > 0$ and $\sum_{k=1}^{K} \pi_k = 1$, and $\mathbf{L}_k \in \mathbb{R}^{p \times r}$ and $\mathbf{R}_k \in \mathbb{R}^{q \times c}$ are the row and column loading matrices with $r \leq p, c \leq q$. Note that $\mathbf{M}_k$, $\mathbf{L}_k$ and $\mathbf{R}_k$ are associated with each component of mixture model. $\mathbf{B}_n^{(k)} \in \mathbb{R}^{r \times c}$ is the latent variable core of $\mathbf{X}_n$ associated with k-th matrix-variate Gaussian component [18, Sec 3.3] with σ_k^2 as residual variance.

Like [3], we introduce a K-dimensional binary random variable $\mathbf{z}$ having a 1-of-K representation in which a particular element z_k is equal to 1 and all other elements are equal to 0. That is, $z_k \in \{0, 1\}$ and $\sum_{k=1}^{K} z_k = 1$. The distribution of $\mathbf{z}$ is defined by

$$p(z_k = 1, z_{k'} = 0, k' \neq k) := \pi_k,$$

which can be written as

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}.$$

Thus the conditional distribution of $\mathbf{X}_n$ given a particular value for $\mathbf{z}_n$ and $\mathbf{B}_n^{(k)}$ is the matrix-variate Gaussian

$$p(\mathbf{X}_n | z_{nk} = 1, \mathbf{B}_n^{(k)}) = \mathcal{N}(\mathbf{X}_n | \mathbf{L}_k \mathbf{B}_n^{(k)} \mathbf{R}_k^T + \mathbf{M}_k, \sigma_k \mathbf{I}, \sigma_k \mathbf{I}).$$

Generally we have

$$p(\mathbf{X}_n | \mathbf{z}_n, \mathbf{B}_n^{(k)}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{X}_n | \mathbf{L}_k \mathbf{B}_n^{(k)} \mathbf{R}_k^T + \mathbf{M}_k, \sigma_k \mathbf{I}, \sigma_k \mathbf{I})^{z_{nk}}.$$

In this model setting, the parameters are $\Theta = \{\pi_k, \mathbf{M}_k, \mathbf{L}_k, \mathbf{R}_k, \sigma_k^2\}(k = 1, .., K)$, and the latent variables are $\mathbf{z}_n$ and $\mathbf{B}_n^{(k)}(n = 1, ..., N)$.

To develop a generative Bayesian model, we define a matrix-variate Gaussian prior $p(\mathbf{B}_n^{(k)})$ over the latent variable with zero-mean unit-covariance, defined as

$$p(\mathbf{B}_n^{(k)}) = \mathcal{N}(0, \mathbf{I}_r, \mathbf{I}_c) = \left(\frac{1}{2\pi}\right)^{\frac{rc}{2}} \cdot \exp\{-\frac{1}{2}\mathbf{tr}(\mathbf{B}_n^{(k)T}\mathbf{B}_n^{(k)})\}.$$

Hence the joint log-likelihood of the observed data set for such a mixture model is:

$$\mathcal{L} = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \ln\{\pi_k p(\mathbf{X}_n, \mathbf{B}_n^{(k)})\}.$$

3. Variational Approximation for mixB2DPPCA Model

We employ the Expectation Maximization (EM) algorithm to solve for model parameters Θ. To maximize the log-likelihood of mixB2DPPCA, we take the expectation of $\mathcal{L}$ with respect to the posterior distribution of both $\mathbf{B}_n^{(k)}$ and z_{nk}, i.e.,

$$\langle\mathcal{L}\rangle = \sum_{n=1}^{N}\sum_{k=1}^{K}\langle z_{nk}\rangle\{\ln\pi_k - \frac{pq}{2}\ln\sigma_k^2 - \frac{1}{2}\mathrm{tr}(\langle\mathbf{B}_n^{(k)T}\mathbf{B}_n^{(k)}\rangle)$$
$$- \frac{1}{2\sigma_k^2}\mathrm{tr}(\mathbf{X}_n - \mathbf{M}_k)^T(\mathbf{X}_n - \mathbf{M}_k)$$
$$+ \frac{1}{\sigma^2}\mathrm{tr}((\mathbf{X}_n - \mathbf{M}_k)^T\mathbf{L}_k\langle\mathbf{B}_n^{(k)}\rangle\mathbf{R}_k^T)$$
$$- \frac{1}{2\sigma^2}\mathrm{tr}(\langle\mathbf{B}_n^{(k)T}\mathbf{L}_k^T\mathbf{L}_k\mathbf{B}_n^{(k)}\rangle\mathbf{R}_k^T\mathbf{R}_k), \quad (2)$$

where $\langle\cdot\rangle$ denotes the expectation.

In E-step, we update Q-distributions of all hidden variables $\mathbf{B}_n^{(k)}$ and z_{nk} with the current fixed parameter values for Θ. In M-step, maximizing the function $\langle\mathcal{L}\rangle$ with respect to the model parameters Θ, we can obtain 'new' values for these parameters.

3.1. Variational E-step

3.1.1 Update the Posterior Distribution of z_{nk}

Suppose $\gamma_{nk} := \langle z_{nk}\rangle$ and it is actually the posterior probability of k-mixture generating data point $\mathbf{X}_n$. By using the same strategy for the mixture Gaussian model [3], we can obtain

$$\gamma_{nk} = \frac{\pi_k p(\mathbf{X}_n|k)}{p(\mathbf{X}_n)}, \quad (3)$$

where $p(\mathbf{X}_n|k)$ is the k-the component, representing the marginal distribution for the observed data $\mathbf{X}_n$ over the latent variable. In our case, the marginal distribution of $\mathbf{X}_n$ is obtained by integrating out the latent variable $\mathbf{B}_n^{(k)}$:

$$p(\mathbf{X}_n|k) = \int p(\mathbf{X}_n|\mathbf{B}_n^{(k)})p(\mathbf{B}_n^{(k)})d\mathbf{B}_n^{(k)}.$$

Different from the vectorial PPCA, we note that the marginal distribution of the observed data $\mathbf{X}_n$ is in general no longer a matrix-variate Gaussian. Thus it is difficult to work with $p(\mathbf{X}_n|k)$ directly. Let $\mathbf{x}_n := \mathrm{vec}(\mathbf{X}_n)$, now we can work with $p(\mathbf{x}_n|k)$ instead of $p(\mathbf{X}_n|k)$. Fortunately, the marginal distribution of $\mathbf{x}_n$ is a multivariate Gaussian distribution when taking the special matrix-variate Gaussian prior $\mathbf{B}_n^{(k)} \sim \mathcal{N}(0, \mathbf{I}_r, \mathbf{I}_c)$. Let $\mathbf{m}_k = \mathrm{vec}(\mathbf{M}_k)$, we can obtain

$$p(\mathbf{x}_n|k) \sim \mathcal{N}(\mathbf{m}_k, \mathbf{C}_k),$$

where the observation covariance model is specified by $\mathbf{C}_k = (\mathbf{R}_k\mathbf{R}_k^T) \otimes (\mathbf{L}_k\mathbf{L}_k^T) + \sigma_k^2\mathbf{I}$. We refer readers to [3, 18] for more details. Then the denominator in (3) becomes

$$p(\mathbf{x}_n) = \sum_{k=1}^{K}\pi_k p(\mathbf{x}_n|k).$$

After getting γ_{nk}, we update the estimated mean matrices $\mathbf{M}_k$'s and mixing proportions π_k's, respectively, by

$$\pi_k = \frac{1}{N}\sum_{n=1}^{N}\gamma_{nk} \quad \text{and} \quad \mathbf{M}_k = \frac{\sum_{n=1}^{N}\gamma_{nk}\mathbf{X}_n}{\sum_{n=1}^{N}\gamma_{nk}}. \quad (4)$$

3.1.2 Update the Posterior Distribution of $\mathbf{B}_n^{(k)}$

In computing the posterior distribution of $\mathbf{B}_n^{(k)}$, we encounter a difficulty as the posteriori distribution of $\mathbf{B}_n^{(k)}$ given $\mathbf{X}_n$

$$p(\mathbf{B}_n^{(k)}|\mathbf{X}_n, \mathbf{L}_k, \mathbf{R}_k, \sigma^2) \propto p(\mathbf{X}_n|\mathbf{B}_n^{(k)}, \mathbf{L}_k, \mathbf{R}_k, \sigma^2)p(\mathbf{B}_n^{(k)})$$

is also in general not a matrix-variate Gaussian. To get a tractable posterior in the variational EM, we restrict the approximated variational distribution to be a matrix-variate Gaussian $\mathcal{N}(\mathbf{B}_n^{(k)}|\mathbf{Q}_n^{(k)}, \mathbf{T}_n^{(k)}, \mathbf{S}_n^{(k)})$ to approximate the true posterior with the mean $\mathbf{Q}_n^{(k)}$ in size $r \times c$ and covariances $\mathbf{T}_n^{(k)} \succ 0$ of size $r \times r$ and $\mathbf{S}_n^{(k)} \succ 0$ of size $c \times c$, respectively. For mixB2DPPCA model, it follows as a natural extension of a single 2DPPCA. So the parameters $\mathbf{Q}_n^{(k)}$, $\mathbf{T}_n^{(k)}$ and $\mathbf{S}_n^{(k)}$ can be estimated through the maximization of a single likelihood function. Particularly, the derived formulas for estimating these parameters are given by, see more details in [26],

$$\mathbf{T}_n^{(k)} = c\sigma_k^2[\mathrm{tr}(\mathbf{R}_k^T\mathbf{R}_k\mathbf{S}_n^{(k)})\mathbf{L}_k^T\mathbf{L}_k + \sigma_k^2\mathrm{tr}(\mathbf{S}_n^{(k)})\mathbf{I}_r]^{-1}$$

$$\mathbf{S}_n^{(k)} = r\sigma_k^2[\mathrm{tr}(\mathbf{L}_k^T\mathbf{L}_k\mathbf{T}_n^{(k)})\mathbf{R}_k^T\mathbf{R}_k + \sigma_k^2\mathrm{tr}(\mathbf{T}_n^{(k)})\mathbf{I}_c]^{-1}$$

and each $\mathbf{Q}_n^{(k)}$ needs to satisfy

$$\mathbf{L}_k^T\mathbf{L}_k\mathbf{Q}_n^{(k)}\mathbf{R}_k^T\mathbf{R}_k + \sigma_k^2\mathbf{Q}_n^{(k)} = \mathbf{L}_k^T(\mathbf{X}_n - \mathbf{M}_k)\mathbf{R}_k.$$

To solve this we need to make a vectorization on both sides and solve a linear equation

$$(\mathbf{R}_k^T \mathbf{R}_k \otimes \mathbf{L}_k^T \mathbf{L}_k + \sigma_k \mathbf{I} \otimes \sigma_k \mathbf{I}) \text{vec}(\mathbf{Q}_n^{(k)}) = \mathbf{y}_n^{(k)} \quad (5)$$

with respect to $\text{vec}(\mathbf{Q}_n^{(k)})$, where

$$\mathbf{y}_n^{(k)} = \text{vec}(\mathbf{L}_k^T (\mathbf{X}_n - \mathbf{M}_k) \mathbf{R}_k),$$

then reshape $\text{vec}(\mathbf{Q}_n^{(k)})$ back to get $\mathbf{Q}_n^{(k)}$.

As we assume the approximated posterior distribution of $\mathbf{B}_n^{(k)}$ is matrix-variate Gaussian, so we can get $\langle \mathbf{B}_n^{(k)} \rangle = \mathbf{Q}_n^{(k)}$ and the following second-order expectations:

$$\langle \mathbf{B}_n^{(k)T} \mathbf{B}_n^{(k)} \rangle = \mathbf{Q}_n^{(k)T} \mathbf{Q}_n^{(k)} + \mathbf{S}_n^{(k)} \text{tr}(\mathbf{T}_n^{(k)}) \quad (6)$$

$$\langle \mathbf{B}_n^{(k)T} \mathbf{L}_k^T \mathbf{L}_k \mathbf{B}_n^{(k)} \rangle = \mathbf{Q}_n^{(k)T} \mathbf{L}_k^T \mathbf{L}_k \mathbf{Q}_n^{(k)} + \mathbf{S}_n^{(k)} \text{tr}(\mathbf{T}_n^{(k)} \mathbf{L}_k^T \mathbf{L}_k) \quad (7)$$

$$\langle \mathbf{B}_n^{(k)} \mathbf{R}_k^T \mathbf{R}_k \mathbf{B}_n^{(k)T} \rangle = \mathbf{Q}_n^{(k)} \mathbf{R}_k^T \mathbf{R}_k \mathbf{Q}_n^{(k)T} + \mathbf{T}_n^{(k)} \text{tr}(\mathbf{S}_n^{(k)} \mathbf{R}_k^T \mathbf{R}_k) \quad (8)$$

3.2. Variational M-step

In the M-step, we fix all the distributions over the hidden variables and gather all the terms containing parameters $\mathbf{L}_k$, $\mathbf{R}_k$ and σ_k^2 in (2) to maximize them respectively. It turns out that:

$$\mathbf{L}_k = [\sum_{n=1}^N \gamma_{nk} (\mathbf{X}_n - \mathbf{M}_k) \mathbf{R}_k \langle \mathbf{B}_n^{(k)} \rangle^T]$$
$$\times [\sum_{n=1}^N \gamma_{nk} \langle \mathbf{B}_n^{(k)} \mathbf{R}_k^T \mathbf{R}_k \mathbf{B}_n^{(k)T} \rangle]^{-1} \quad (9)$$

$$\mathbf{R}_k = [\sum_{n=1}^N \gamma_{nk} (\mathbf{X}_n - \mathbf{M}_k)^T \mathbf{L}_k \langle \mathbf{B}_n^{(k)} \rangle]$$
$$\times [\sum_{n=1}^N \gamma_{nk} \langle \mathbf{B}_n^{(k)T} \mathbf{L}_k^T \mathbf{L}_k \mathbf{B}_n^{(k)} \rangle]^{-1} \quad (10)$$

and

$$\sigma_k^2 = \frac{1}{pqN_k} \{ \sum_{n=1}^N \gamma_{nk} \text{tr}(\mathbf{X}_n - \mathbf{M}_k)^T (\mathbf{X}_n - \mathbf{M}_k)$$
$$- 2 \sum_{n=1}^N \gamma_{nk} \text{tr}(\mathbf{R}_k \langle \mathbf{B}_n^{(k)} \rangle^T \mathbf{L}_k^T (\mathbf{X}_n - \mathbf{M}_k)$$
$$+ \sum_{n=1}^N \gamma_{nk} \text{tr}(\langle \mathbf{B}_n^{(k)T} \mathbf{L}_k^T \mathbf{L}_k \mathbf{B}_n^{(k)} \rangle \mathbf{R}_k^T \mathbf{R}_k) \} \quad (11)$$

where $N_k = \sum_n \gamma_{nk}$.

The overall variational EM algorithm is to alternate between E-step and M-step. The final variational EM algorithm is summarized in Algorithm 1.

Algorithm 1 Variational EM algorithm for mixB2DPPCA.

Initialize: Training set $\mathcal{X} = \{\mathbf{X}_n\}_{n=1}^N$; Initialize all of model parameters Θ and covariance matrices $\mathbf{T}_n^{(k)}$ and $\mathbf{S}_n^{(k)}$, $n = 1, ..., N$ and $k = 1, ..., K$.

1: **for** $t = 1$ to T **do**
2:    **Variational E-step:**

- Iterate the mean matrix $\mathbf{Q}_n^{(k)}$ based on (5) and update the second-order expectations based on (6), (7) and (8).

- Update each γ_{nk}, mixing proportions π_k and mean matrices $\mathbf{M}_k$ based on (3) and (4).

3:    **Variational M-step:**

- Maximize objective function $\langle \mathcal{L} \rangle$ with respect to each elements $\mathbf{L}_k$, $\mathbf{R}_k$ and σ_k^2 based on (9), (10) and (11).

4: **end for**

Define the average reconstruction error

$$\mathbf{e}(t) = \sqrt{\frac{\sum_{n=1}^N \|\mathbf{X}_n - \widehat{\mathbf{X}}_n^{(t)}\|_F^2}{N}} \quad (12)$$

where $\widehat{\mathbf{X}}_n = \mathbf{L}_{k'} \mathbf{B}_n^{(k')} \mathbf{R}_{k'}^T + \mathbf{M}_{k'}$ with $k' = \arg\max_k \{\gamma_{nk}\}$ the reconstructed image.

Algorithm 1 may terminate either a given maximum iterative number T is achieved or the following condition is satisfied,

$$|\mathbf{e}(t) - \mathbf{e}(t+1)| \leq \epsilon \quad (13)$$

where ϵ is a given error tolerance.

3.3. The Reduced-Dimensionality Representation for a New Sample

In order to obtain the reduced-dimensionality representation for a given sample, we should solve for the latent variable cores. From the probabilistic perspective, the posterior mean $\mathbf{Q}_{new}^{(k)} := \langle \mathbf{B}_{new}^{(k)} | \mathbf{X}_{new} \rangle$ can be seen as the reduced-dimensionality representation, which is a $r \times c$ feature matrix and given by solving a linear equation

$$(\mathbf{R}_k^T \mathbf{R}_k \otimes \mathbf{L}_k^T \mathbf{L}_k + \sigma_k \mathbf{I} \otimes \sigma_k \mathbf{I}) \text{vec}(\mathbf{Q}_{new}^{(k)}) = \mathbf{y}_{new}^{(k)}$$

with respect to $\text{vec}(\mathbf{Q}_{new}^{(k)})$, where

$$\mathbf{y}_{new}^{(k)} = \text{vec}(\mathbf{L}_k^T (\mathbf{X}_{new} - \mathbf{M}_k) \mathbf{R}_k),$$

then reshape $\text{vec}(\mathbf{Q}_{new}^{(k)})$ back to get $\mathbf{Q}_{new}^{(k)}$. At the same time, we can compute the corresponding $\gamma_{new,k}$, i.e., the

posterior probability of k-th component generating the new sample, given by

$$\gamma_{new,k} = \frac{p(\mathbf{X}_{new}|k)\pi_k}{p(\mathbf{X}_{new})}.$$

We find the largest $\gamma_{new,k}$ ($k = 1, ..., K$) from which the most appropriate local 2DPPCA model can be identified for the new sample. That is, a natural choice is to assign the new sample to a cluster with the largest posterior probability.

4. Experimental Results and Analysis

In this section, we conduct several experiments on some public databases to assess the proposed mixB2DPPCA model. These experiments are designed to evaluate the performance of the proposed mix2DPPCA in reconstruction and recognition by comparing with existing models and algorithms.

The relevant PCA algorithms that can be fairly compared against our proposed mixB2DPPCA are GLRAM (Generalized Low Rank Approximations of Matrices) [25], PSOPCA (Probabilistic Second-Order PCA) [27], mixture of PPCA [20] with the code from http://www.science.uva.nl/~jverbeek. Because the zero-noise PSOPCA model and GLRAM have the same stationary point [27], we only compare with GLRAM.

4.1. Data Preparation and Experiment Setting

All of the experiments are conducted on the following four public available datasets:

- A subset of handwritten digits images from the M-NIST database (http://yann.lecun.com/exdb/mnist).

- The Yale face database (http://vision.ucsd.edu/content/yale-face-database).

- The AR face database (http://rvl1.ecn.purdue.edu/aleix/aleix_face_DB.html).

- The FERET face database (http://www.itl.nist.gov/iad/humanid/feret/feret_master.html).

The subset of handwritten digits images is selected from MNIST database, which contains 1000 digital images with 100 images of each digit. All images are in grayscale and have a uniform size of 28×28 pixels.

The Yale face database contains 15 individuals, with 11 images for each individual. The images were captured under different illumination and expression conditions. The images are all 100×100 pixels with 256 grey levels. In the experiments, we randomly select 6 images of each person as the training samples, and use the remaining images to form

the testing sample set. All images are scaled to a resolution of 64×64 pixels.

The AR face database contains over 4,000 color images corresponding to 126 subjects. There are variations of facial expressions, illumination conditions, and occlusions (sun glasses and scarf) with each person. Each individual consists of 26 frontal view images taken in two sessions (separated by 2 weeks), where each session has 13 images. Figure 1 shows the 26 images of one subject. In the experiments, we select 30 subjects (15 man and 15 women), and only use the non-occluded 14 images (i.e., the first seven face images of each row in Figure 1). The first seven of each subject are used for training and the last seven for testing. All images are cropped and resized to 50×40 pixels.

FERET database includes 1400 images of 200 different subjects, with 7 images per subject. In the experiments, we select 50 subjects randomly. Five images of each subject are used for training and the remained images are used for testing. All images are cropped and resized to 32×32 pixels.

In experiments, the initial mixing proportions are set to $\pi_k = 1/K$ and the initial loading matrices $\mathbf{L}_k$ and $\mathbf{R}_k$ are given randomly. Besides, we choose randomly K samples as mean matrices $\mathbf{M}_k$ of the mixture gaussian model and set all $\sigma_k^2 = 1$.

4.2. Reconstruction Performance

In this section, we test reconstruction error of the proposed mixB2DPPCA model (1). Applying the proposed model, all digital images can be softly grouped into K clusters, each of which is modelled by a local B2DPPCA. From all the trained γ_{nk}, the most appropriate local B2DPPCA for a given sample can be found. Then we use the most appropriate local B2DPPCA to reconstruct the initial digit image, that is:

$$\widehat{\mathbf{X}}_n = \mathbf{L}_{k'} * \mathbf{Q}_n^{(k')} * \mathbf{R}_{k'}^T + \mathbf{M}_{k'},$$

where k' represents the k'-th local B2DPPCA which most appropriate to the sample $\mathbf{X}_n$. After obtaining all reconstructed digit images $\widehat{\mathbf{X}}_n$, we can using the equation (12) to compute the average reconstruction error.

Next we compare the reconstruction error of different algorithms on three databases. In all algorithms, we set the iterative number is $T = 50$ and the reduced dimension is $r = c = 4$.

4.2.1 Reconstruction Error on Digit Image Set

We use the given digital image subset in Section 4.1 as training set. In this phase, we compare the reconstruction error of the training set.

Figure 2 shows the average reconstruction error of the relevant algorithms. From left to right, the component num-
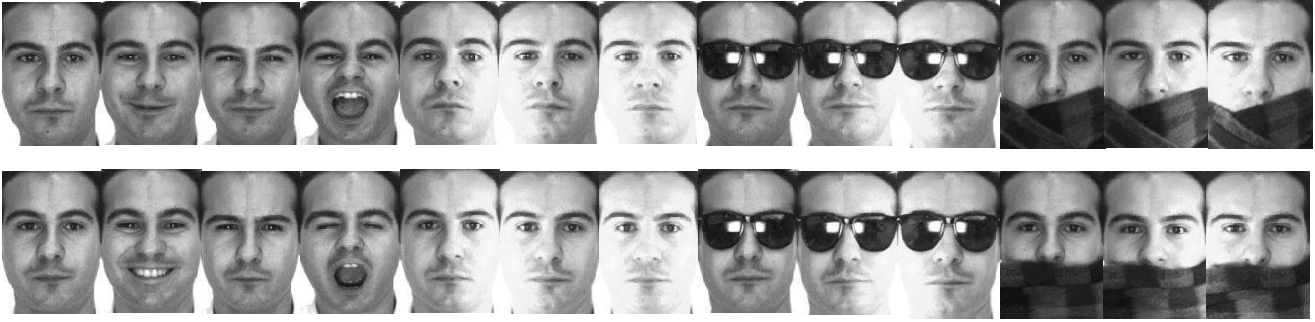
Figure 1: Twenty-six face examples of one subject from AR database. The first row is from the first session, and the second row images are from the second session.
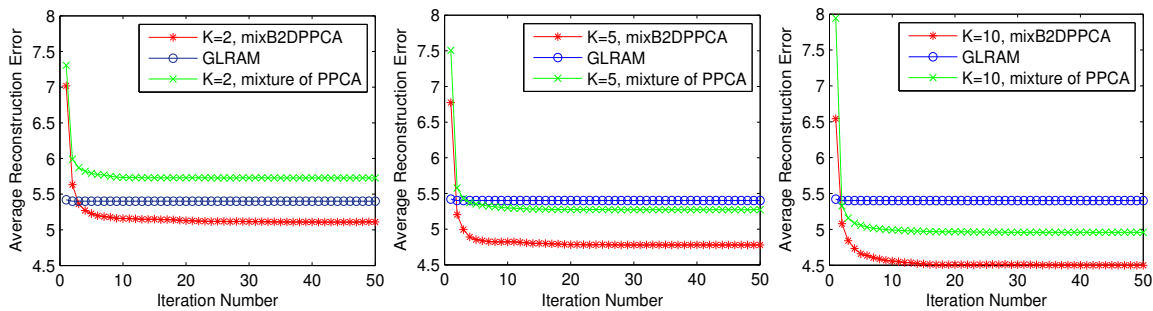

Figure 2: Average reconstruction error versus iteration number with the components number $K = 2$, $K = 5$ and $K = 10$ from the left to right.

ber is $K = 2$, $K = 5$ and $K = 10$ respectively. First-ly, from these three sub-figures, we can see that the reconstruction error of GLRAM algorithm has no change. This is because GLRAM has no relationship with K. Besides, GLRAM works by iteratively computing the leading eigenvectors of the left and right one-sided sample covariance matrices. Thus GLRAM convergent in five steps and the change of reconstruction error is not obvious in the figure. Secondly, fixing the same number of reduced dimension, the performance of our proposed mixB2DPPCA is better than GLRAM. From the view of compression, decoded images from our algorithm have higher quality for the compression ratio of $49 : 1$. It illustrates that mixB2DPPCA can correctly identify data according to clusters. When K becomes larger, the mixB2DPPCA outperform the mixture of PPCA in terms of reconstruction errors.

The reconstructed images of different methods are shown in Fig. 3 with $K = 10$. The first row shows four original images. The second, third and fourth rows are the reconstructed images by GLRAM, mixture of PPCA and mixB2DPPCA, respectively. It can be found that the proposed mixB2DPPCA has better reconstruction outcomes, while the results of other two methods show a little degra-

dation.

4.2.2 Reconstruction Error on Yale and AR Databases

In this experiment, we compare the reconstruction error on Yale and AR databases.

Figure 4 shows the average reconstruction error of all the algorithms: (a) on the Yale database and (b) on the AR database. The component number is $K = 5$ and the reduced dimensionality is $(r, c) = (4, 4)$. It is obvious that the reconstruction error of mixB2DPPCA on testing set has reduced greatly than other algorithms.

Figure 5 shows some reconstructed images of different algorithms on Yale database. The first row is four original images. The last three rows are the corresponding images reconstructed by mixture of PPCA, GLRAM and mixB2DPPCA. It can be shown that the results of our algorithm have better visual effect than that of GLRAM. Besides we can also see that although the face images reconstructed by mixture of PPCA are relatively clear, they don't match the same original images visually. The reconstructed images on AR database are shown in Figure 6. The first row shows five original images in the test set and the last three

r,c	GLRAM	$K=4$		$K=6$		$K=8$	
		mixPPCA	mixB2DPPCA	mixPPCA	mixB2DPPCA	mixPPCA	mixB2DPPCA
2	0.6133	0.5760±0.0401	0.6400±0.0267	0.6237±0.0219	**0.6720±0.0210**	0.6519±0.0124	0.6693±0.0250
4	0.7067	0.6376±0.0222	0.7173±0.0197	0.6642±0.0245	0.7146±0.0180	0.6613±0.0201	**0.7200±0.0089**
6	0.7200	0.6480±0.0289	0.7200±0.0154	0.6506±0.0186	0.7187±0.0203	0.6480±0.0283	**0.7320±0.0160**
8	0.7200	0.6786±0.0117	0.7240±0.0227	0.6560±0.0265	0.7187±0.0262	0.6640±0.0233	**0.7347±0.0160**

Table 1: Recognition accuracy of GLRAM, mixture of PPCA and mixB2DPPCA training on the Yale database

r,c	GLRAM	$K=6$		$K=8$		$K=10$	
		mixPPCA	mixB2DPPCA	mixPPCA	mixB2DPPCA	mixPPCA	mixB2DPPCA
4	0.5714	0.5328±0.0220	0.6671±0.0333	0.5595±0.0214	0.7000±0.0371	0.5752±0.0297	**0.7244±0.0381**
6	0.6857	0.6252±0.0242	0.7867±0.0138	0.6343±0.0236	**0.8017±0.0291**	0.6613±0.0182	0.7576±0.0366
8	0.7190	0.7004±0.0190	0.8116±0.0231	0.7100±0.0246	0.8211±0.0246	0.7133±0.0222	**0.8357±0.0237**

Table 2: Recognition accuracy of GLRAM, mixture of PPCA and mixB2DPPCA training on the AR database
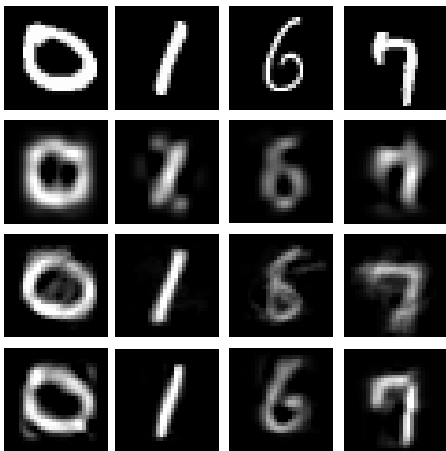

Figure 3: Original images and reconstructed images: The first row shows four original digital images. The second, third and fourth rows are the reconstructed images by GLRAM, mixture of PPCA and mixB2DPPCA, respectively.

Figure 4: Average reconstruction error versus iteration number with the components number $K = 5$ on Yale database (a) and AR database (b).

rows are the reconstructed images from three models.

From the reconstruction experiments, we can conclude that mixB2DPPCA generally outperforms global linear 2D-PCA algorithms in terms of reconstruction errors. It demonstrates that the classification of training set in advanced is important for the performance of feature extraction.

4.3. Recognition Performance

In this section, we compare the recognition performances of GLRAM, mixture of PPCA and mixB2DPCA on Yale, AR and FERET face databases. These algorithms can be used for extracting features of facial images from the training samples, respectively, and then a nearest neighbor classifier (1-NN) is used to find the most-similar face from the training samples for a querying face. In our exper-
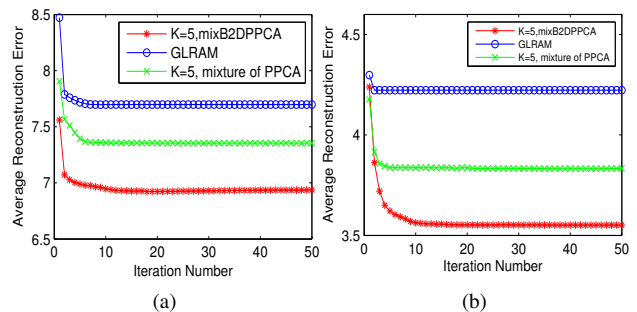
iments, the distance measure between two sets of feature matrices $\mathbf{B}_{n_1}$ and $\mathbf{B}_{n_2}$, is defined as

$$\text{dist} = \sum_{k=1}^{K} \|\mathbf{B}_{n_1}^{(k)} - \mathbf{B}_{n_2}^{(k)}\|_F.$$

where $\mathbf{B}_n = [\mathbf{B}_n^{(1)}, ..., \mathbf{B}_n^{(K)}]$ represents the combination of K latent variable cores related with n-th sample[1]. In all algorithms, we set maximum iteration number is 50 and ϵ is 1E-3. We repeat the procedure 10 times, and the mean values and relevant variances are reported in Tables 1 to 3.

Table 1 shows the recognition rates of three feature extraction algorithms: GLRAM, mixture of PPCA and mixB2DPPCA training on Yale database. The mean values and relevant variances are reported for the cases of the reduced dimension $(r,c) = (2,2)$, $(4,4)$, $(6,6)$ and $(8,8)$. For the mixture of PPCA and mixB2DPPCA, we also computed the recognition rates for the different component number K ($K = 4,6,8$), shown in Table 1. First-

---

[1] A more accurate way is to use $\gamma_{n_1 k}\gamma_{n_2 k}$ to weight the individual distance.

r, c	GLRAM	$K = 6$		$K = 8$		$K = 10$	
		mixPPCA	mixB2DPPCA	mixPPCA	mixB2DPPCA	mixPPCA	mixB2DPPCA
4	0.5000	0.4620±0.0315	0.6070±0.0427	0.4690±0.0470	**0.6210±0.0326**	0.4840±0.0316	0.5900±0.0429
6	0.5300	0.5140±0.0206	**0.6733±0.0541**	0.5350±0.0283	**0.6467±0.0343**	0.5320±0.0297	0.6644±0.0328
8	0.5400	0.5440±0.0298	0.6900±0.0458	0.5610±0.0159	**0.6945±0.0526**	0.5580±0.0187	**0.6770±0.0593**
10	0.5500	0.5910±0.0460	0.6890±0.0455	0.5720±0.0364	0.6960±0.0599	0.5970±0.0336	**0.7100±0.0573**

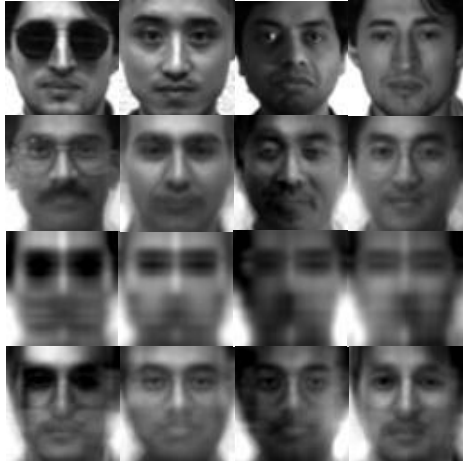Table 3: Recognition accuracy of GLRAM, mixture of PPCA and mixB2DPPCA training on the FERET database

Figure 5: Original and reconstructed images (in the Yale database): The first row is original images. The second, third and fourth rows are the reconstructed images by mixture PPCA, GLRAM and mixB2DPPCA, respectively.

Figure 6: Original and reconstructed images (in the AR database): The first row is original images. The second, third and fourth rows are the reconstructed images by GLRAM, mixture PPCA and mixB2DPPCA, respectively.

ly, from the table we can see that the recognition rates of the mixture of PPCA and mixB2DPPCA have a little fluc-

tuation compared with GLRAM. This may be caused by the uncertainty of probability. Secondly, compared with GLRAM, the mean recognition rates of mixB2DPPCA algorithm have obviously improved. The bold figures are the best results in the comparison.

Table 2 shows the recognition rates of the above three algorithms training on AR database. The reduced dimensions are $(r, c) = (4, 4)$, $(6, 6)$ and $(8, 8)$ and component numbers are $K = 6, 8, 10$, respectively. From the table we can see that the mean recognition rates of mixB2DPPCA algorithm have better improvement over the other two algorithms.

Table 3 shows the recognition rates on FERET database. The reduced dimensions are $(r, c) = (4, 4)$, $(6, 6)$, $(8, 8)$ and $(10, 10)$, and the component numbers are $K = 6, 8, 10$, respectively. In this case, both the mixture of PPCA and the proposed mixB2DPPCA produce slightly larger variances, however the mean recognition rates have risen greatly. GLRAM is relatively more robust.

5. Conclusions

In this paper, we proposed a mixture of bilateral-projection probabilistic PCA model for feature extraction and dimensionality reduction. Different from the standard PCA which is a global dimension reduction model, this model employs the mixture of matrix-variate Gaussian to model local linear sub-models. All the parameters in the resulting probabilistic model can be estimated through the maximization of the likelihood function. The new model not only makes good use of spatial (structural) information of 2D data but also can softly group data into a given number of clusters. The performance of feature extraction of the proposed method generally outperforms other existing 2D algorithms in terms of reconstruction error and recognition rate. The approach used in this paper can be readily extended to higher order tensorial data and other non-Gaussian noise models can also be integrated into the model such.

6. Acknowledgement

References

[1] C. Archambeau. Probabilistic models in noisy environments and their application to a visual prosthesis for the blind. In *Unpublished doctoral dissertation, Université Catholique de Louvain, Belgium*, 2005.

[2] C. Archambeau, N. Delannay, and M. Verleysen. Robust probabilistic projections. In *International Conference on Machine Learning*, pages 33–40, 2006.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[4] C. M. Bishop and M. E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):281–293, 1998.

[5] J. Gao. Robust L1 principal component analysis and its Bayesian variational inference. *Neural Computation*, 20(2):555–572, 2007.

[6] J. Gao and R. Xu. Mixture of the robust L1 distributions and its applications. In *Lecture Notes in Artificial Intelligence*, volume 4830, pages 26–35, 2007.

[7] Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. In *Technical Report CRG-TR-96-1, University of Toronto*, 1996.

[8] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using Laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[9] P. Hoyer and A. Hyvärinen. Independent component analysis applied to feature extraction from colour and stereo images. *Network: Computation in Neural Systems*, 11(3):191–210, 2000.

[10] F. Ju, Y. Sun, J. Gao, Y. Hu, and B. Yin. Image outlier detection and feature extraction via L1-norm based 2D probabilistic PCA. *IEEE Transactions on Image Processing*, 24(12):4834–4846, 2015.

[11] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition*, pages 506–513, 2004.

[12] H. C. Kim, D. Kim, and S. Y. Bang. An efficient model order selection for PCA mixture model. *Pattern Recognition Letters*, 24(9):1385–1393, 2003.

[13] H. Kriegel, P. Kröger, and E. Schubert. A general framework for increasing the robustness of PCA-based correlation clustering algorithms. In *Scientific and Statistical Database Management. Springer Berlin Heidelberg*, pages 418–435, 2008.

[14] J. Lu, K. Plataniotis, A. Venetsanopoulos, and S. Li. Ensemble-based discriminant learning with boosting for face recognition. *IEEE Transactions on Neural Network*, 17(1):166–178, 2006.

[15] D. Peel and G. McLachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, 10:339–348, 2000.

[16] D. Ridder and V. Franc. Robust subspace mixture models using t-distributions. In *Proceedings of the 14th British Machine Vision Conference*, pages 1–10, 2003.

[17] T. Su and J. G. Dy. Automated hierarchical mixtures of probabilistic principal component analyzers. In *International Conference on Machine learning, ACM*, 2004.

[18] N. H. Timm. *Applied Multivariate Analysis*. Springer, 2002.

[19] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[20] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999.

[21] H. Wang, S. Chen, Z. Hu, and B. Luo. Probabilistic two-dimensional principal component analysis and its mixture model for face recognition. *Neural Computing and Applications*, 17(5-6):541–547, 2008.

[22] H. Wang and J. Wang. 2DPCA with L1-norm for simultaneously robust and sparse modelling. *Neural Network*, 46(10):190–198, 2013.

[23] X. Xie, S. Yan, J. Kwok, and T. Huang. Matrix-variate factor analysis and its applications. *IEEE Transactions on Neural Networks*, 19(10):1821–1826, 2008.

[24] J. Yang, D. Zhang, A. F. Frangi, and J. Yang. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[25] J. Ye. Generalized low rank approximations of matrices. *Machine Learning*, 61(1-3):167–191, 2005.

[26] S. Yu, J. Bi, and J. Ye. Matrix-variate factor analysis and its applications. In *KDD Workshop Data Mining Using Matrix and Tensors*, pages 1–7, 2008.

[27] S. Yu, J. Bi, and J. Ye. Matrix-variate and higher-order probabilistic projections. *Data Mining and Knowledge Discovery*, 22(3):372–392, 2011.