An EM Based Probabilistic Two-Dimensional CCA with Application to Face Recognition

Mehran Safayani · Seyed Hashem Ahmadi · Homayun Afrabandpey · Abdolreza Mirzaei

Abstract Recently, two-dimensional canonical correlation analysis (2DCCA) has been successfully applied for image feature extraction. The method instead of concatenating the columns of the images to the one-dimensional vectors, directly works with two-dimensional image matrices. Although 2DCCA works well in different recognition tasks, it lacks a probabilistic interpretation. In this paper, we present a probabilistic framework for 2DCCA called probabilistic 2DCCA (P2DCCA) and an iterative EM based algorithm for optimizing the parameters. Experimental results on synthetic and real data demonstrate superior performance in loading factor estimation for P2DCCA compared to 2DCCA. For real data, three subsets of AR face database and also the UMIST face database confirm the robustness of the proposed algorithm in face recognition tasks with different illumination conditions, facial expressions, poses and occlusions.

1 Introduction

Although many real-world applications encounter high dimensional data, the most informative part of the data can be modeled in a low dimensional space. Moreover, processing high-dimensional data is a time consuming process and requires lots of resources. To tackle these problems, feature extraction has been used as a tool for finding a compact and meaningful data representation.

Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, IRAN
Tel.: +98 31 33919063
Fax: +98 31 33912451
E-mail: safayani@cc.iut.ac.ir(Corresponding Author) · hashem.ahmadi@ec.iut.ac.ir · h.afraei@ec.iut.ac.ir · mirzaei@cc.iut.ac.ir

For single-mode source data, some subspace learning methods are conducted to learn more semantic description subspaces. Examples of these methods are principal component analysis (PCA) [1] and linear discriminant analysis (LDA). However, for observations from two sources that share some mutual information, canonical correlation analysis (CCA) [2] is a very popular approach for dimensionality reduction. CCA seeks a lower-dimensional space where two sets of variables are maximally correlated after projecting on it. This technique is widely used in different fields of pattern recognition, computer vision, bioinformatics, etc. [3,4,5]. In the CCA-based methods, it is necessary to vectorize 2D image matrices. Vectorization has three main drawbacks: (**I**) breaking the spatial structure of image data which may cause losing potentially useful structural information among column/rows [6], (**II**) leading to a high-dimensional vector space and small sample size problem which in turn makes it difficult to calculate the covariance matrices [7] and (**III**) causing the covariance matrices to be very large which in turn makes the eigen-decomposition of such large matrices very time-consuming.

To overcome these drawbacks, in 2007 two-dimensional CCA (2DCCA) was introduced by Lee and Choi [8] which computes CCA directions based on 2D image matrices. The proposed 2DCCA overcomes the curse of dimensionality and significantly reduces the computational cost, by directly working with 2D images instead of reshaping them into 1D vectors. In [8], higher recognition accuracies were reported using 2DCCA compared to CCA using two face databases and the time complexity has been improved.

However, an associated probabilistic model for observed data was notably absent form these feature extraction methods. A probabilistic feature extraction algorithm could be intuitively appealing for so many reasons [9]. To bridge the gap, in 1999, Tipping and Bishop proposed probabilistic PCA [9] based on a latent variable model known as factor analysis (FA) [10,11]. The proposed PPCA was then used as a framework for many other new formulations for PCA [12,13,14,15]. Also, there have been some probabilistic models proposed for LDA [16,17]. In 2005, Bach and Jordan [18] also proposed a probabilistic interpretation of CCA and estimate the parameters of their proposed model using both maximum likelihood and expectation maximization. Recently, many inspiring research proceeded in the 1D CCA domain, including kernel based, semiparametric and nonparametric methods [19,20,21], but in the 2D CCA domain, we feel that more work is required. To bridge the gap, a probabilistic model of 2DCCA was introduced by Safayani et al. in 2011 [22]. They showed that the maximum likelihood estimation of parameters, leads to the two dimensional canonical correlation directions. However, they didn't propose an EM based solution for their model. EM does not require the explicit eigen-decomposition of covariance matrices. Moreover, using EM it is possible to handle models with incomplete data such as mixture models where the cluster labels are the missing values [23].

In this paper, we present a probabilistic interpretation of 2DCCA, referred to as P2DCCA, together with an EM based solution to estimate the parameters of the model. The proposed model can handle the small sample size problem

effectively.

The rest of the paper is organized as follows: Section 2 briefly reviews some related algorithms such CCA, PCCA and 2DCCA which are necessary to understand how the proposed algorithms work. The proposed P2DCCA model is introduced in Section 3. In Section 4, some experiments on synthetic data and several face databases are given to evaluate performance of the proposed algorithm; finally, the paper is concluded in Section 5.

2 Background

2.1 Canonical Correlation Analysis (CCA)

Imagine that we are given two sets of random vectors t_1 and t_2 where $t_{1,n} \in \mathrm{R}^{D_1}$ and $t_{2,n} \in \mathrm{R}^{D_2}$ for $n \in 1, 2, ..., N$ are realizations of the corresponding random vectors, respectively. CCA seeks transformation vectors $w_1 \in \mathrm{R}^{D_1}$ and $w_2 \in \mathrm{R}^{D_2}$ such that correlation between $w_1^T t_1$ and $w_2^T t_2$ are maximized. The correlation between $w_1^T t_1$ and $w_2^T t_2$ can be formulated as

$$\rho = \frac{cov(w_1^T t_1, w_2^T t_2)}{\sqrt[2]{var(w_1^T t_1)var(w_2^T t_2)}} = \frac{w_1^T \Sigma_{12} w_2}{\sqrt[2]{(w_1^T \Sigma_{11} w_1)(w_2^T \Sigma_{22} w_2)}} \tag{1}$$

where $\Sigma_{ij} = \frac{1}{N} \sum_{n=1}^{N} (t_{i,n} - \mu_i)(t_{j,n} - \mu_j)^T$ for $i, j \in 1, 2$ is the cross-covariance matrix of t_1 and t_2 and $\mu_i = \frac{1}{N} \sum_{n=1}^{N} t_{i,n}$ for $i \in \{1, 2\}$ denotes the mean vector of t_i.

Then, the objective function for CCA can be written as:

$$\operatorname*{argmax}_{w_1, w_2} w_1^T \Sigma_{12} w_2 \tag{2}$$
$$s.t. \ \ w_1^T \Sigma_{11} w_2 = 1$$
$$w_2^T \Sigma_{22} w_2 = 1$$

Optimizing such a constrained maximization problem with respect to w_1 and w_2 leads to the following generalized eigenvalue problem:

$$\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \lambda \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{3}$$

By solving equation (3), w_1 and w_2 that maximize the correlation between the projected data can be found.

2.2 Probabilistic CCA

The generative model introduced by Bach and Jordan for CCA is as follows:

$$t_i = W_i z + \mu_i + \epsilon_i \qquad i \in \{1, 2\} \tag{4}$$

In this model, $W_i \in \mathrm{R}^{D_i \times d}$ $i \in \{1,2\}$ are linear projections that map two sets of high dimensional observed random vectors $t_i \in \mathrm{R}^{D_i}, i \in \{1,2\}$ to a set of lower dimensional latent vectors $z \in \mathrm{R}^d$. Here μ_i is the mean vector for x_i and ϵ_i is the error term which is assumed to follow a multivariate Gaussian distributions with zero mean and inverse covariance matrix Ψ_i. Bach and Jordan proved that the maximum likelihood estimation for the parameters of this model would lead to the canonical directions. The maximum likelihood estimates of the projection matrices are given by:

$$\widehat{W_i} = \Sigma_{ii} U_{id} M_i, \quad i \in \{1,2\} \tag{5}$$

where Σ_{ii} is the sample covariance matrix, U_{id} are the first d canonical directions and $M_i \in \mathrm{R}^{d \times d}$ where $i \in \{1,2\}$ are arbitrary matrices such that $M_1 M_2 = P_d$, where P_d is the diagonal matrix of the first d canonical correlations. Figure 1 shows a graphical representation of the model.
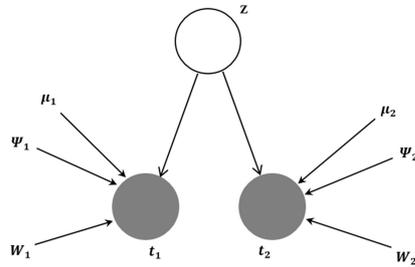

Fig. 1: Graphical model for probabilistic CCA

2.3 Two-Dimensional CCA

Two-dimensional CCA (2DCCA) was proposed to tackle the problem of vectorizing data in CCA. For each random matrix T_1 and T_2, 2DCCA introduces left transforms u_i and right transforms v_i where $i \in \{1,2\}$. After the projection, data would have the form $u_i^T T_i v_i$. 2DCCA finds these left and right transforms in a way to maximize the correlation between projected data. Therefore, the objective function of 2DCCA can be formulated as:

$$\underset{u_1,u_2,v_1,v_2}{\mathrm{argmax}} \ cov(u_1^T T_1 v_1, u_2^T T_2 v_2) \tag{6}$$
$$s.t. \ var(u_1^T T_1 v_1) = 1$$
$$var(u_2^T T_2 v_2) = 1,$$

u_1 and u_2 can be obtained by solving the generalized eigenvalue problem (7) with fixed v_1 and v_2:

$$\begin{bmatrix} 0 & \Sigma_{12}^r \\ \Sigma_{21}^r & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \lambda \begin{bmatrix} \Sigma_{11}^r & 0 \\ 0 & \Sigma_{22}^r \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{7}$$

In a similar way, given u_1 and u_2, right transforms v_1 and v_2 can be found by solving

$$\begin{bmatrix} 0 & \Sigma_{12}^l \\ \Sigma_{21}^l & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \lambda \begin{bmatrix} \Sigma_{11}^l & 0 \\ 0 & \Sigma_{22}^l \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{8}$$

where $\Sigma_{i,j}^r$, $i,j \in \{1,2\}$ is the cross-covariance matrix between T_i and T_j and Σ_{ii}^r, $i \in \{1,2\}$ is the auto-covariance matrix of T_i defined as follows, respectively:

$$\Sigma_{ij}^r = \frac{1}{N} \sum_{n=1}^N (T_{i,n} - \mu_i) v_i v_j^T (T_{j,n} - \mu_j)^T, \qquad i,j \in \{1,2\} \tag{9}$$

$$\Sigma_{ij}^l = \frac{1}{N} \sum_{n=1}^N (T_{i,n} - \mu_i) u_i u_j^T (T_{j,n} - \mu_j)^T, \qquad i,j \in \{1,2\} \tag{10}$$

where $T_{i,n}$ for $n \in \{1, ..., N\}$ is the realization of the random matrix T_i and $\mu_i = \frac{1}{N} \sum_{n=1}^N T_{i,n}$ is the corresponding matrix of mean values.
Left transforms (l_x and l_y) and right transforms (r_x and r_y) are obtained by iteratively solving equations (7) and (8), until convergence. The eigenvectors associated with d_1 largest eigenvalues in (7) determine left transform matrices U_1 and U_2 and the eigenvectors associated with d_2 largest eigenvalues in (8) determine right transform matrices V_1 and V_2. Using these transform matrices it is possible to project data from a high dimensional space to a new lower dimensional feature space.

3 Probabilistic Two Dimensional CCA (P2DCCA)

In this section, we propose probabilistic two-dimensional CCA and an EM-based solution for finding the parameters of the model. In our model, observed data are modeled as two-dimensional matrices as follows:

$$T_i = U_i Z V_i^T + \mu_i + \Xi_i \quad i \in \{1,2\} \tag{11}$$

where $T_i \in \mathrm{R}^{m_i \times n_i}$ for $i \in \{1,2\}$ are observed matrices and $Z \in \mathrm{R}^{m' \times n'}$ is the latent matrix. $U_i \in \mathrm{R}^{m_i \times m'}$, $V_i \in \mathrm{R}^{n_i \times n'}$ are projection matrices, μ_i is the mean matrix of the observed data and Ξ_i is the residual matrix. Based on this definition, parameters of the model are $\{U_i, V_i\}_{i=1}^2$ and the parameters of the distribution of Ξ_i'.

Let $D_i = \{T_{i;n}\}_{n=1}^N$ where $i \in \{1, 2\}$ be a set containing N observed data matrices and $\{Z_n\}_{n=1}^N$ be the corresponding latent variable set. Then the complete data would be $(T_{1,n}, T_{2,n}, Z_n)$ and the log likelihood of the complete data can be written as

$$L_c = \sum_{n=1}^N log \ p(T_{1,n}, T_{2,n}, Z_n) \tag{12}$$

To estimate the parameters, first we must calculate expectation of the log-likelihood and then take the derivative of the expected log-likelihood with respect to each parameter. Unfortunately there is no closed-form solution for computing the projection matrices $\{U_i, V_i\}_{i=1}^2$ simultaneously. Inspired by [24], a decoupled probabilistic model is employed to obtain projection matrices separately using an alternating optimization procedure. In such a model, we first assume that the value of one set of projection matrices, e.g. the right projections $\{V_i\}_{i=1}^2$, is known. Then observations are projected to the corresponding latent spaces. The projection procedure is a probabilistic one that introduced in section (3.3). By doing this, the left probabilistic model is defined as

$$T_i^l = U_i Z^l + \Xi_i^l, \quad i = 1, 2 \tag{13}$$

where Z^l is the left model latent matrix, μ_i^l is the mean matrix of the left projected observations and Ξ_i^l is the noise source for left probabilistic model where columns of the noise matrix follow a normal distribution with zero mean and covariance matrix Ψ_i^l. By such definition, parameter set for the left probabilistic model would be $\Theta^l = \{U_i, \Psi_i^l\}_{i=1}^2$ which can be estimated using expectation maximization procedure. The estimation procedure is explained later in this section.

In a similar procedure and parallel to the left probabilistic model, for the right probabilistic model, we assume that the left projection matrices, i.e. $\{U_i\}_{i=1}^2$, are known. Then observations are projected over the corresponding latent spaces,hence the right probabilistic model is defined as

$$T_i^r = V_i Z^r + \Xi_i^r, \quad i = 1, 2 \tag{14}$$

Similar to the left probabilistic model, Z^r, μ_i^r and Ξ_i^r are defined for the right model where in this model the noise source have $N(0, \Psi_i^r)$ distribution. The parameter set for the right probabilistic model would be $\Theta^r = \{V_i, \Psi_i^r\}_{i=1}^2$. Using these definitions, the decoupled predictive density $p(T_1, T_2, Z)$ could be defined as

$$p(T_1, T_2, Z) \propto p(T_1^l, T_2^l, Z^l)p(T_1^r, T_2^r, Z^r) \tag{15}$$

Now we can rewrite the log likelihood of equation (15) as

$$L_c = \sum_{n=1}^N log(p(T_{1,n}^l, T_{2,n}^l, Z_n^l)p(T_{1,n}^r, T_{2,n}^r, Z_n^r)) \tag{16}$$

$$= \sum_{n=1}^N log \ p(T_{1,n}^l, T_{2,n}^l, Z_n^l) + \sum_{n=1}^N log \ p(T_{1,n}^r, T_{2,n}^r, Z_n^r).$$

To apply the EM algorithm to the decoupled probabilistic model, in E-step expectation of log likelihood for the left probabilistic model and the right probabilistic model is computed, separately. Then each of the expected log likelihood is maximized with respect to its parameters. In the following subsections we describe how to optimize left and right probabilistic model respectively.

3.1 Optimizing the left probabilistic model

Let $t_{i,j}^l$ be the jth column vector of $T_i^l \in R^{m_i \times n'}$. By assuming columns of T^l to be independent of each other, the distribution of T_i^l is defined as

$$p(T_i^l) = \prod_{j=1}^{n'} p(t_{i,j}^l), \qquad i \in \{1, 2\} \tag{17}$$

We also consider $z_j^l \in R^{m' \times 1}$ as the jth column vector of Z^l which has normal distribution of $N(0, I)$ and also in the same way $\mu_{i,j}^l$ is the jth column vector of μ_j^l. Based on equation (13) and the distribution considered for Z^l and Ξ_i^l, it can be concluded that

$$p(t_{i,j}^l) \sim N(\mu_{i,j}^l, U_i U_i^T + \Psi_i^l), \qquad i = 1, 2 \tag{18}$$

Suppose $\tau_{n,j}^l = [(t_{1,n,j}^l)^T \ (t_{2,n,j}^l)^T]^T \in R^{(m_1+m_2) \times 1}$, $V = [U_1^T \ U_2^T]^T \in R^{(m_1+m_2) \times m'}$, $m_j^l = [(\mu_{1,j}^l)^T \ (\mu_{2,j}^l)^T]^T \in R^{(m_1+m_2) \times 1}$ and $\Psi^l = \begin{pmatrix} \Psi_1^l & 0 \\ 0 & \Psi_2^l \end{pmatrix}$ for the left probabilistic model, where $t_{i,n,j}^l$ refers to the jth column vector of nth image in the ith observation set where $i \in \{1, 2\}$. Therefore, distributions of $p(\tau_j^l)$ can be obtained as follows:

$$p(\tau_j^l) \sim N(m_j^l, \Sigma^l), \qquad j \in [1, n'] \tag{19}$$

where $\Sigma^l = U U^T + \Psi^l$ and we assume $\Sigma^l > 0$.
Based on (17) we can write:

$$p(T_{1,n}^l, T_{2,n}^l, Z_n^l) = \prod_{j=1}^{n'} p(\tau_{n,j}^l, z_{n,j}^l) = \prod_{j=1}^{n'} p(\tau_{n,j}^l | z_{n,j}^l) p(z_{n,j}^l) \tag{20}$$

To apply the EM algorithm to the decoupled probabilistic model, for each of the probabilistic models expectation of the log likelihood function is calculated in the E-step $E(L_c^l)$ where the detail is given in the appendix, and then maximization step (M-step) is done by maximizing $E(L_c^l)$ with respect to V and Ψ^l. By doing so, the values of the parameters are estimated as

$$U_{t+1} = \frac{\widetilde{\Sigma}^l (\Psi_t^l)^{-1} U_t (M_t^l)^{-1}}{(M_t^l)^{-1} + (M_t^l)^{-1} U_t^l (\Psi_t^l)^{-1} \widetilde{\Sigma}^l (\Psi_t^l)^{-1} U_t (M_t^l)^{-1}} \tag{21}$$

$$\Psi_{t+1}^l = \begin{pmatrix} (\widetilde{\Sigma}^l - \widetilde{\Sigma}^l(\Psi_t^l)^{-1}U_t(M_t^l)^{-1}U_{t+1}^T)_{11} & 0 \\ 0 & (\widetilde{\Sigma}^l - \widetilde{\Sigma}^l(\Psi_t^l)^{-1}U_t(M_t^l)^{-1}U_{t+1}^T)_{22} \end{pmatrix} \quad (22)$$

where $M^l = I + U^T(\Psi^l)^{-1}U$ and A_t shows the value of parameter A in iteration t and $\widetilde{\Sigma}^l$ is the sample covariance matrix of observed data for the left probabilistic model , i.e.

$$\widetilde{\Sigma}^l = \frac{1}{N}\sum_{n=1}^{N}[(T_{1,n}^l - \mu_1^l)^T \ (T_{2,n}^l - \mu_2^l)^T]^T[(T_{1,n}^l - \mu_1^l)^T \ (T_{2,n}^l - \mu_2^l)^T] \quad (23)$$

3.2 Optimizing the right probabilistic model

In the manner similar to optimization of the left probabilistic model we have:

$$p(T_i^r) = \prod_{j=1}^{m'}p(t_{i,j}^r), \qquad i \in \{1,2\} \quad (24)$$

where $t_{i,j}^r$ is the jth column vector of $T_i^r \in \mathrm{R}^{n_i \times m'}$. Then $p(t_{i,j}^r)$ is computed as:

$$p(t_{i,j}^r) \sim N(\mu_{i,j}^r, V_iV_i^T + \Psi_i^r), \qquad i \in \{1,2\} \quad (25)$$

Let $\tau_{n,j}^r = [(t_{1,n,j}^r)^T \ (t_{2,n,j}^r)^T]^T \in \mathrm{R}^{(n_1+n_2)\times 1}$, $V = [V_1^T \ V_2^T]^T \in \mathrm{R}^{(n_1+n_2)\times n'}$, $m_j^r = [(\mu_{1,j}^r)^T \ (\mu_{2,j}^r)^T]^T \in \mathrm{R}^{(n_1+n_2)\times 1}$ and $\Psi^r = \begin{pmatrix} \Psi_1^r & 0 \\ 0 & \Psi_2^r \end{pmatrix}$, where $t_{i,n,j}^r$ refers to the jth column vector of nth image in the ith observation set. Then $p(\tau_j^r)$ and $p(T_{1,n}^r, T_{2,n}^r, Z_n^r)$ are obtained as follows:

$$p(\tau_j^r) \sim N(m_j^r, \Sigma^r), \qquad j \in [1, m'] \quad (26)$$

$$p(T_{1,n}^r, T_{2,n}^r, Z_n^r) = \prod_{j=1}^{m'}p(\tau_{n,j}^r, z_{n,j}^r) = \prod_{j=1}^{m'}p(\tau_{n,j}^r|z_{n,j}^r)p(z_{n,j}^r) \quad (27)$$

where $\Sigma^r = VV^T + \Psi^r > 0$. Given the details in appendix (A), after computing $E(L_c^r)$ in the E-step, the parameters V and Ψ^r are computed by maximizing the likelihood in the M-step. So, we have:

$$V_{t+1} = \frac{\widetilde{\Sigma}^r(\Psi_t^r)^{-1}V_t(M_t^r)^{-1}}{(M_t^r)^{-1} + (M_t^r)^{-1}V_t^r(\Psi_t^r)^{-1}\widetilde{\Sigma}^r(\Psi_t^r)^{-1}V_t(M_t^r)^{-1}} \quad (28)$$

$$\Psi_{t+1}^r = \begin{pmatrix} (\widetilde{\Sigma}^r - \widetilde{\Sigma}^r(\Psi_t^r)^{-1}V_t(M_t^r)^{-1}V_{t+1}^T)_{11} & 0 \\ 0 & (\widetilde{\Sigma}^r - \widetilde{\Sigma}^r(\Psi_t^r)^{-1}V_t(M_t^r)^{-1}V_{t+1}^T)_{22} \end{pmatrix} \quad (29)$$

where $M^r = I + V^T(\Psi^r)^{-1}V$ and $\widetilde{\Sigma}^r$ is computed as follows:

$$\widetilde{\Sigma}^r = \frac{1}{N}\sum_{n=1}^{N}[(T_{1,n}^r - \mu_1^r)^T \ (T_{2,n}^r - \mu_2^r)^T]^T[(T_{1,n}^r - \mu_1^r)^T \ (T_{2,n}^r - \mu_2^r)^T] \quad (30)$$

3.3 Probabilistic projection and dimension reduction

We can project the observation matrices into the latent space using the standard projection matrices ,i.e., $\{U_1, U_2, V_1, V_2\}$. However, as described in [9], it is more natural to use probabilistic projections. In this regard, we represent each projected observation matrix, T_i, by mean of distribution of corresponding latent space, i.e., $E(Z|T_i)$. For the left model, it can be shown that

$$E(Z^l|T_1, T_2) = (M^r)^{-1} \begin{bmatrix} V_1^T & V_2^T \end{bmatrix} \begin{bmatrix} \Psi_1 & 0 \\ 0 & \Psi_2 \end{bmatrix}^{-1} \begin{bmatrix} T_1 - \mu_1 \\ T_2 - \mu_2 \end{bmatrix} \qquad (31)$$

$E(Z^l|T_1)$ and $E(Z^l|T_2)$ are obtained by marginalizing (31) over T_2 and T_1 respectively. So we have

$$E(Z^l|T_i) = (M^r)^{-1} V_i^T (\Psi_i^r)^{-1} (T_i - \mu_i), \qquad i \in \{1, 2\} \qquad (32)$$

Similarly for the right model we have

$$E(Z^r|T_i) = (M^l)^{-1} U_i^T (\Psi_i^l)^{-1} (T_i - \mu_i), \qquad i \in \{1, 2\} \qquad (33)$$

The procedure for dimension reduction is given by sequential projection in left and right models as

$$E(Z|T_i) = (M^l)^{-1} U_i^T (\Psi_i^l)^{-1} (T_i - \mu_i)((M^r)^{-1} V_i^T (\Psi_i^r)^{-1})^T \qquad (34)$$

The P2DCCA algorithm is summarized in Figure 2. The proposed P2DCCA model benefits the ability to extend to other methods such as Mixtures of P2DCCA, Bayesian P2DCCA and also to robust P2DCCA.

4 Experimental Results

We evaluated our algorithm on both synthetic and real data. In synthetic data part we verified our implementation of P2DCCA algorithm by simple synthetic data and randomly generated projected matrices. we also compared our algorithm with 2DCCA method in projection matrices estimation . For real part evaluation, the proposed P2DCCA method was used for face recognition on two well-known face image databases (AR [25] and UMIST [26]). The AR database is divided into three subsets for evaluating the performance of the system in regard to different illumination, expression and occlusion conditions. The UMIST database is used to obtain the performance in dealing with pose variation.

1   **Input**: the matrices $T_{i,n}|_{n=1}^N \in \Re^{m_i \times n_i}$, $i \in \{1,2\}$

2   **Output**: Find $U_i \in \Re^{m_i \times m'}$, $V_i \in \Re^{n_i \times n'}$, Ψ^l and Ψ^r

3   **Initialize**: $V_{1,i}^l$ and $U_{1,i}^l$ using 2DCCA and initialize Ψ_1^l and Ψ_1^r with I

4   *for t=1,...,T_{max}*

5         $M^r = I + V_t^T (\Psi_t^r)^{-1} V_t$

6         $T_{i,n}^l = (M^r)^{-1} V_{t,i}^T (\Psi_{t,i}^r)^{-1} (T_{i,n} - \mu_i)$, $i \in \{1,2\}$, $n = 1,...,N$

7   *Initialize* Ψ_t^l randomly

8   while L_c^l not converge

9   *E-step:*

10   *Compute* $M^l = I + U_t^T (\Psi_t^l)^{-1} U_t$

11   *Compute* $E\{z_{n,j}^l\} = (M^l)^{-1} U_t^T (\Psi_t^l)^{-1} (\tau_{n,j}^l - m_j^l)$

12   *Compute* $E\{(z_{n,j}^l)^T z_{n,j}^l\} = (M^l)^{-1} + E\{z_{n,j}^l\} E\{z_{n,j}^l\}^T$

13   *M-step:*

14             $U_{t+1} = \frac{\Sigma_{n=1}^N \Sigma_{j=1}^{n'} (\tau_{n,j}^l - m_j^l) E\{z_{n,j}^l\}}{\Sigma_{n=1}^N \Sigma_{j=1}^{n'} E\{z_{n,j}^l (z_{n,j}^l)^T\}}$

15             $A^l = \frac{1}{n'} \Sigma_{n=1}^N \Sigma_{j=1}^{n'} (\tau_{n,j}^l - m_j^l)(\tau_{n,j}^l - m_j^l)^T + U_{t+1} E\{x_{n,j}^l\}(\tau_{n,j}^l - m_j^l)$

16             $\Psi_{t+1}^l = \begin{pmatrix} A_{11}^l & 0 \\ 0 & A_{22}^l \end{pmatrix}$

17   End while

18        $M^l = I + U_t^T (\Psi_t^l)^{-1} U_t$

19        $T_{i,n}^r = (M_t^l)^{-1} U_{t,i}^T (\Psi_{t,i}^l)^{-1} (T_{i,n} - \mu_i)$, $i \in \{1,2\}$, $n = 1,...,N$

20   *Initialize* Ψ_t^r randomly

21   while L_c^r not converge

22   *E-step:*

23   *Compute* $M^r = I + V_t^T (\Psi^r)^{-1} V_t$

24   *Compute* $E\{z_{n,j}^r\} = (M^r)^{-1} V_t^T (\Psi^r)^{-1} (\tau_{n,j}^r - m_j^r)$

25   *Compute* $E\{(z_{n,j}^r)^T z_{n,j}^r\} = (M^r)^{-1} + E\{z_{n,j}^r\} E\{z_{n,j}^r\}^T$

26   *M-step:*

27             $V_{t+1} = \frac{\Sigma_{n=1}^N \Sigma_{j=1}^{m'} (\tau_{n,j}^r - m_j^r) E\{z_{n,j}^r\}}{\Sigma_{n=1}^N \Sigma_{j=1}^{m'} E\{z_{n,j}^r (z_{n,j}^r)^T\}}$

28             $A^r = \frac{1}{m'} \Sigma_{n=1}^N \Sigma_{j=1}^{m'} (\tau_{n,j}^r - m_j^r)(\tau_{n,j}^r - m_j^r)^T + V_{t+1} E\{z_{n,j}^r\}(\tau_{n,j}^r - m_j^r)$

29             $\Psi_{t+1}^r = \begin{pmatrix} A_{11}^r & 0 \\ 0 & A_{22}^r \end{pmatrix}$

30   End while

31   *end;*

Fig. 2: Procedure of P2DCCA.

4.1 Experiments on synthetic data

In this section, we aim to verify our implementation of the proposed method with the simplest possible scenario. So we generate some synthetic data and projection matrices. Then estimate the projection matrices using our method and compare them with the true ones. We know that P2DCCA estimations are up to rotation, hence to simplify comparison we assume Z to be 1×1 dimension. We set dimensions of T_1 and T_2 to 5×5. Then we generate 1000 samples of Z from a normal distribution with zero mean and unit variance, also we randomly generate the elements of $U_i \in \Re^5$, $V_i \in \Re^5$, $i \in \{1,2\}$ using

a uniform distribution in [0, 1] interval and consider them as the ground truth projection matrices, then T_i will obtained using (11). In this equation, each element of the residual matrices sampled from a gaussian distribution with 0 mean and σ_i^2 variance. Having the synthetic data we run P2DCCA algorithm as discussed in Figure 2 and calculate U_i and V_i. We also run 2DCCA and obtain the corresponding projection matrices. Then we compare obtained matrices by these two algorithms with the ground truth projection matrices. To cancel the scale factors we divide each transform by its norm before comparison. Euclidean distance is utilized to compare the normalized transforms. Figure 3 shows the results. It is obvious that in the worst-case the distance value becomes one and in the best-case it is zero. As it is depicted in this figure, P2DCCA estimation of U_i and V_i for $i \in \{1, 2\}$ are much closer to the ground truth compared to the those obtained by 2DCCA. In this experiment we used 1000 generated samples and set $\sigma_i = 0.1$. To examine the effect of these selections, we repeat our experiment with different values of sample numbers and also noise variances. Figure 4 demonstrates the results. As it can be observed from this figure the P2DCCA estimation in all cases is much closer to the ground truth compared to the 2DCCA method.
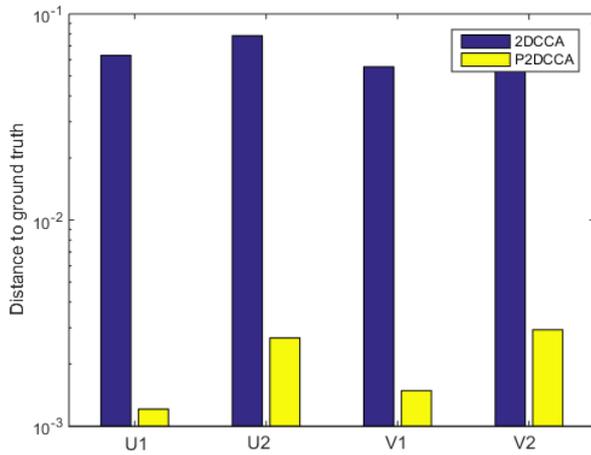

Fig. 3: Comparison of 2DCCA and P2DCCA in loading factors estimation on the synthetic data with 1000 generated samples and $\sigma = 0.1$

4.2 Experiments on the AR database

The AR face database contains over 4,000 color face images including frontal views of faces with different facial expressions, illumination conditions and
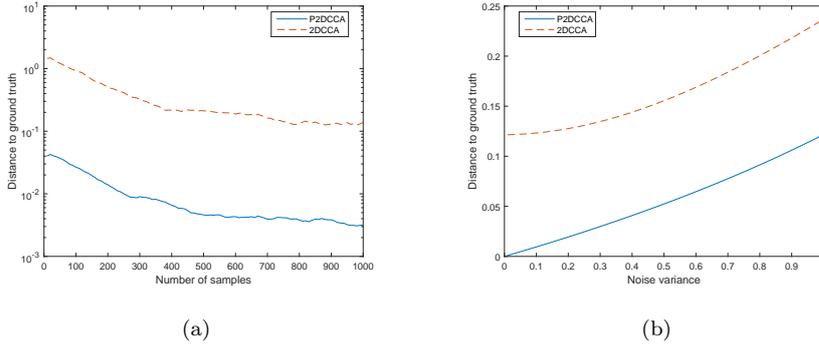
(a)   (b)

Fig. 4: Comparison of 2DCCA and P2DCCA in U1 loading factor estimation on the synthetic data

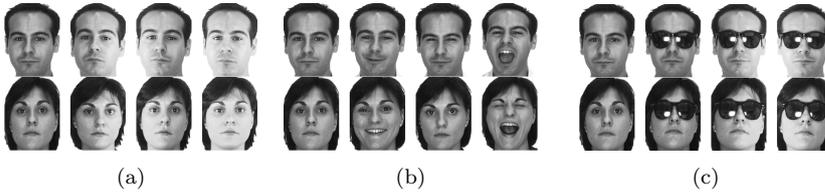(a)   (b)   (c)

Fig. 5: Exemplary face images in (a). AR-1, (b). AR-2 and (c). AR-3

occlusions. For most individuals, there are two sessions of images which were taken in two different time periods. Each session contains 13 images. In our experiments, we used the first session, because some individuals do not have the second session of images. We collected 1310 face images of 131 people (72 male and 59 female). For each person, there are 10 different face images in our collected images: three with different illumination conditions; three with different expressions; three with occlusions and the remaining images are those with neutral expression and no occlusion which are known as reference images in our experiments. To examine the performance of the proposed methods in different conditions, we partitioned the collected images into three subsets known as AR-1, AR-2 and AR-3. For each individual, AR-1 contains four images, three of which are images with different lighting conditions and the remaining one is the reference image. AR-2 is used to test performance of the algorithms when there exist expression variation. AR-2 involves four images per individuals: three images have different expressions and the last one is the reference image. AR-3 is prepared to test performance in the presence of occlusion. Again, this subset contains four images per individuals where three images were taken with glasses and the last one is the reference image. Figure 5 shows exemplary face images of a man and a woman in AR-1, AR-2 and AR-3, respectively. Image are gray scaled, resized and then normalized to 50×50 pixels. We compared the performance of the proposed method with a range of different supervised and unsupervised dimensionality reduction al-

gorithms and different versions of them including PCA, LDA, CCA, PPCA, PCCA, 2DPCA, 2DLDA and 2DCCA. Both PCA and LDA based methods work with one set of data. Also LDA based methods are supervised while PCA and CCA based algorithms are unsupervised. The task here is to investigate how well different algorithms can relate face images with varying illumination conditions, expressions and occlusion, in correspondence to the reference face images. The CCA, PCA, LDA, PPCA, PCCA, 2DCCA, 2DPCA, 2DLDA and P2DCCA are used to extract features from facial images and then a 1-NN classifier is employed for classification. Note that based on output type of each of the algorithms (vector or matrix), for 2DCCA, 2DPCA, 2DLDA and P2DCCA, Frobenius distance is used to calculate the distance between two feature matrices, while for CCA, PCA, LDA, PPCA and PCCA the common Euclidean distance measure is adopted. Furthermore, it should be noted here that since PCCA suffers the small sample size problem, implementing it using formulas introduced in [18] caused the covariance matrices to be singular. To solve the problem, we did dimension reduction using PCA before implementing the algorithm.

To evaluate the recognition accuracy, we used "three-fold cross-validation". As it is evident, CCA based algorithms need two sets of images for training, where in this paper the training sets are called left training set and Right training set. To form the training sets, e.g., for AR-1, neutral images (images with no illumination) are considered as Left training set, while to form the Right training set, one of the three images of each individual with different illumination conditions is selected randomly. The other two images are considered as test images. This procedure is repeated for three times, where each time a different image among the three images is selected for the Right training set, while neutral images are always used to form the Left training set. To test the performance of each algorithm, all images of both right and left training sets are projected on the new feature spaces using their corresponding transforms. Also, each of the test images is projected on both feature spaces, so we have two projection for every test image. Then we calculate the distance between each of the two projected test images and projected training images. The label of the training image with the nearest projection to any of the two test image projection determine the final class of the test image. This procedure iterates until we find the final class for all images in the test set. These final classes are compared to the real classes of the images and the recognition accuracy of each algorithm is calculated. Finally, the average recognition rate of the three round experiments is recorded as the final recognition accuracy.

Since PCA and LDA based algorithms work with one set of data, to have a fair comparison we used two images as training and the other two images as the test data, where neutral images are always in the training data together with one of the other images with different illuminations in each iterations. Again the process is repeated three times and the final accuracy is the average of the three runs. Figure 7 shows the test process for P2DCCA.

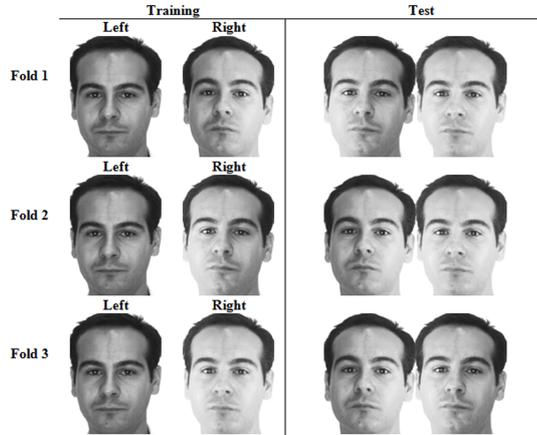Train and test procedure for AR2 and AR3 subsets are similar to AR1 and Ta-

Fig. 6: A representation of training and testing images used in each fold for AR-1

ble 1 through Table 3 demonstrate the recognition accuracy of evaluated algorithms for the experiments conducted on AR-1, AR-2 and AR-3, respectively. In these tables, d is the dimension of the reduced feature space. Note that output for two dimensional algorithms (2DCCA, 2DPCA, 2DLDA, P2DCCA and MP2DCCA) is of matrix type with dimension $d \times d$, while for CCA, PCA, LDA, PPCA and PCCA methods output is a vector of dimension d. In these results we see that P2DCCA get the best performance among all tested methods. We see about 10% improvement of recognition rate for P2DCCA over 2DCCA in AR-1 and AR-3, and about 3% improvement in AR-2.

Table 1: Comparison of the average recognition accuracy rates of the nine evaluated algorithms on AR-1 (%)

		d = 5	d = 10	d = 15	d = 20	d = 25	d = 30	d = 35	d = 40	d = 45	d = 50	d = 130	Best
	PCA	28.6	34.2	38.5	41.9	46.3	48.9	49.7	50.6	51.8	52.5	**56.9**	**56.9**
	LDA	19.3	33	40.1	44.7	46.6	48.5	50	50.3	50.9	**51.3**		**51.3**
	CCA	6.5	16.5	30.2	38.2	49.4	57.3	63.3	63	68	71	**87.8**	**87.8**
	PPCA	29.5	34.6	39.1	42.2	46.3	49.2	50.3	50.9	51.8	52.7	**56.9**	**56.9**
AR1	PCCA	13.4	28	44	54.2	61.1	65	67.7	72	73.3	76.9	**88**	**88**
	2DPCA	49.1	56.7	56.7	57	57	56.9	**57.3**	57.1	57.1	57.1		**57.3**
	2DLDA	22.5	44.1	47.5	49.1	50.6	51.7	52.3	53	53.4	**66.3**		**66.3**
	2DCCA	35	72.1	74.9	83.1	82.2	**85.9**	84.6	84.4	83.8	83.4		**85.9**
	P2DCCA	55.5	88.2	93.2	93.4	94.3	94.4	94.1	93.5	**94.5**	93.5		**94.5**

Figure 8 shows how the log-likelihoods of the left probabilistic model and the right probabilistic model of P2DCCA improve with each iteration. As it can be seen in the figure, both left and right models converge.

It should be noted that it is very common that in the algorithms for optimizing row and column projections only one iteration with the iterative algo-

Table 2: Comparison of the average recognition accuracy rates of the nine evaluated algorithms on AR-2 (%)

		d = 5	d = 10	d = 15	d = 20	d = 25	d = 30	d = 35	d = 40	d = 45	d = 50	d = 130	Best
	PCA	41.7	54.1	58.5	61.3	63	63.6	64.6	65.5	66	67.2	**67.6**	**67.2**
	LDA	51	66.5	72.6	75.6	78.5	80.2	80.7	81.2	**81.3**	81.3		**81.3**
	CCA	5.1	12.6	21.4	29.6	34.9	40.8	45.4	48.5	50.9	55.2	**76.7**	**76.7**
	PPCA	42.6	54.6	59	61.5	63.5	64.1	64.8	65.9	66.4	66.9	**67.7**	**67.7**
AR_2	PCCA	18.1	30.2	42.5	47	55.1	56	58.4	61.2	63	66.4	**77.2**	**77.2**
	2DPCA	65.5	68.2	**68.4**	67.8	67.9	67.7	67.7	67.4	67.7	67.7		**68.4**
	2DLDA	65.3	**71.6**	70.6	69.6	69.6	69.3	69.5	69	69	69.3		**71.6**
	2DCCA	63.4	80.8	**81.7**	79	79	79.5	79.5	78.8	78.6	79.1		**81.7**
	P2DCCA	66.4	80.4	82.3	82.2	82.8	83	83.6	83.5	83.3	**84**		**84**

Table 3: Comparison of the average recognition accuracy rates of the nine evaluated algorithms on AR-3 (%)

|		d = 5	d = 10	d =15	d = 20	d = 25	d = 30	d = 35	d = 40	d = 45	d = 50	d = 130	Best
	PCA	14.9	33.2	41.5	47	51.1	54.5	57	58	58	59.2	**61.7**	**61.7**
	LDA	25.4	41.1	48.1	53.6	55.1	56.5	57.3	57.9	58.2	**58.8**		**58.8**
	CCA	5.7	12.5	23.9	32.3	38.5	43.4	50.3	55.3	60.4	63.2	**80.1**	**80.1**
	PPCA	18.7	35.1	43.1	47.7	51.7	54.7	57.3	58.1	58.5	59.5	**62.5**	**62.5**
AR_3	PCCA	11.6	23.3	39.2	44.7	54.1	60.1	63.7	63.9	70.7	72	**81**	**81**
	2DPCA	55.1	60.9	62.2	62.9	63.4	63.5	**63.6**	63.4	63.1	63.1		**63.6**
	2DLDA	33.8	41.9	49.9	51.7	51.8	52.4	53.3	51.7	52.3	**62.3**		**62.3**
	2DCCA	41.2	57.5	66.2	73	73.5	73.3	76.6	74.4	74.8	**76.6**		**76.6**
	P2DCCA	57.6	79.6	82.3	84.2	84.5	83.6	84.6	84.9	**85.4**	84.6		**85.4**
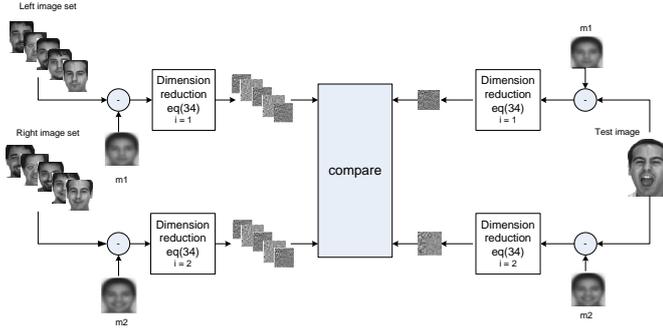
Fig. 7: Graphical representation of recognition process in P2DCCA

rithm is performed[27]. Therefore in all the experiments we use one iteration of the algorithm, i.e., $T_{max} = 1$. This significantly reduces computational cost of the algorithm. We tried more iterations and got no significant improvement in the recognition rate. Figure 9 shows the results for 1 to 5 iterations for AR1, AR2 and AR3. This results support the idea of choosing $T_{max} = 1$.

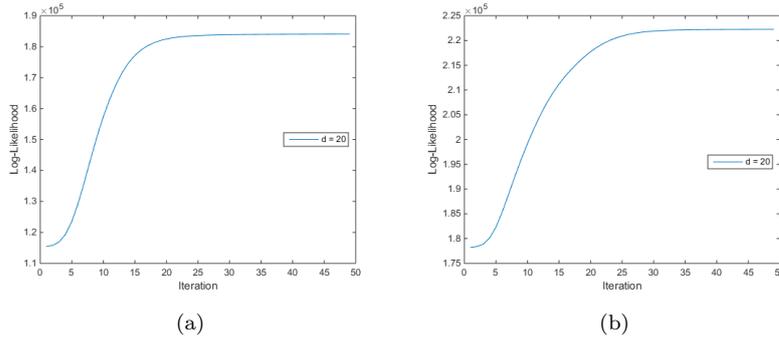(a)   (b)

Fig. 8: Log-Likelihood values versus the number of iterations for (a) left probabilistic model and (b) Right probabilistic model of P2DCCA for d = 20
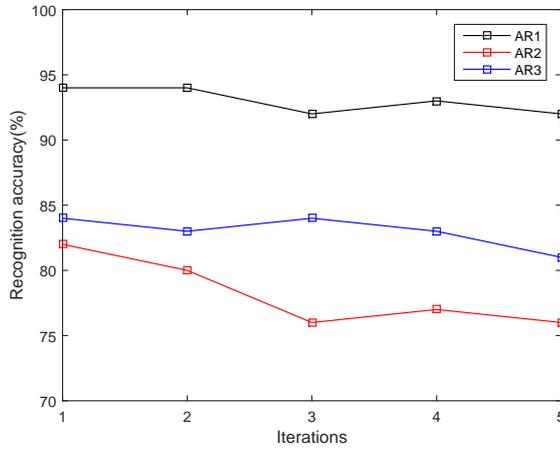

Fig. 9: Recognition rate of AR1, AR2 and AR3 for $Tmax > 1$

4.3 Experiments on the UMIST database

The UMIST face database also known as Sheffield face database [28] consists of 564 images of 20 subjects. Subjects have different races, sexes, and appearances. For each subject, there are images with different poses from profile to frontal view. Images have 256 grey levels with resolution of 220×220 pixels. In our experiment, 360 images with 18 samples per subject are used to examine performance of different algorithms when face orientation varies significantly. Figure 10 shows 18 images of one subject.

We select frontal image as well as seven other randomly selected images for training set and the remaining images for the test set. In the training phase of CCA based methods, frontal image always is selected as the left training

Fig. 10: Eighteen face examples of one subject from the UMIST database

image and one of the seven other images as the right training image. 1-NN classifier is used for classification. This procedure is repeated for twenty times, and the average recognition rates of algorithms are reported. Table 5 shows the recognition accuracy of evaluated algorithms for the experiments conducted on UMIST. In this test while P2DCCA achieved slightly better performance compared to 2DCCA, it is not the best. In fact LDA achieved the best performance. In the AR test, LDA had 2 images per class for training, but here it has 8 images per class which leaded to best performance for this supervised method. Ignoring LDA, we see that P2DCCA performance is higher than other methods.

Table 4: Comparison of the average recognition accuracy rates of the nine evaluated algorithms on UMIST (%)

		d = 5	d = 10	d = 15	d = 20	d = 25	d = 30	d = 35	d = 40	d = 45	d = 50	d = 75	Best
	PCA	46.4	48.8	50.9	52.1	55.6	56.7	55.1	55.4	57.3	56.2	57.4	57.4
	LDA	95.5	97.9	98.9									98.9
	CCA	23.3	46.5	61.9	75.2	79.9	84.7	83.6	87.3	87.2	91.7	92.8	92.8
	PPCA	43.5	50.8	54.9	54.5	56.5	57	59	56.3	55	58	57	59
UMIST	PCCA	51	77.8	86.4	87.1	90.4	90.1	90.2	94.9	93.4	93.6	94.2	94.9
	2DPCA	57.2	57.6	58.3	57.5	56.5	56.2	55.7	55.4	57	57.7		58.3
	2DLDA	70.2	80.1	79.8	81.2	82.9	78.5	80.5	79.3	82.2	77.8		82.9
	2DCCA	83.1	93.6	96.1	96.5	97.3	97	97.5	97	97.2	97.2		97.5
	P2DCCA	81.6	95.3	97.5	97.5	97.4	97.7	98	98	98.2	98.3		98.3

Since the UMIST face dataset contains 20 subjects to be discriminated, the LDA features is limited to 19. To be able to compare the results of LDA with that of the other algorithms, we showed the results for $d = 5$, $d = 10$ and $d = 15$ for LDA and larger values of d for other algorithms.

4.4 Evaluation of the Experimental Results

The above experiments showed that the accuracy of P2DCCA is consistently better than other CCA based methods, i.e. CCA, PCCA and 2DCCA. But, a question sill remains: "Are these differences statistically significant?". In this section we answered the question by evaluating the experimental results using independent-samples T-test (or independent t-test, for short). In this section and also the next section, we only considered CCA based algorithms including

CCA, PCCA, 2DCCA and P2DCCA since the goal of this paper is to compare the functionality of the newly proposed CCA based method with that of the other CCA based algorithms. The desired significance level is 0.05 and the null hypothesis is that there is no significant difference between recognition rates of P2DCCA and CCA, PCCA and 2DCCA, respectively. We reject the null hypothesis whenever the resulted ρ-value becomes lower than 0.05 and in this case the result can be considered statistically significant. It is necessary to note that to run the t-test in each dataset, for each algorithm we considered the highest recognition rate. Table 5 shows the ρ-value of the test. As can be seen from this table, P2DCCA significantly outperforms other algorithms and the null-hypothesis has been rejected in all cases.

Table 5: The ρ-value associated with the null hypothesis: "no significant difference between recognition rates of P2DCCA and the corresponding algorithm"

Data Sets \Algorithms	CCA	PCCA	2DCCA
AR-1	1.9e-06	9.5e-07	8.6e-10
AR-2	6.1e-12	5.7e-12	7.5e-05
AR-3	2.5e-05	4.8e-05	5.2e-06
UMIST	6.1e-05	3.5e-05	5.7e-03

4.5 Computational Complexity

This section compares the computational cost of the algorithms. To compare the time complexity of the algorithms, we consider input images of size $m \times m$ where we want to reduce their dimension to $d \times d$ in case of two-dimensional algorithms. For CCA and PCCA, vectorization caused the input data to have dimension $m^2 \times 1$ and the output to have dimension $d \times 1$. However, for simplicity, d is considered to be equal to m, i.e. $d = m$ in our analysis. Table 6 shows the computational complexity of the algorithms. In this table, N is the number of random samples in the dataset. It should be noted that there are two types of iteration in the corresponding methods; one is the iteration necessary for the convergence of the EM part of the algorithm and the other is the iteration for alternating the optimization procedure between left and right model. We show the former by t and the latter by r in the table. However, $r = 1$ in our experiments.

Table 6: Time complexity of algorithms

Algorithm	CCA	PCCA	2DCCA	P2DCCA
Complexity	$O(m^6 + Nm^4)$	$O(tm^6 + Nm^4)$	$O(rm^3 + rNm^3)$	$O(trm^3 + rNm^3)$

5 Conclusion

This paper proposed a probabilistic model for two dimensional CCA termed as P2DCCA together with an EM-based solution to estimate the parameters of the model. Experimental results demonstrated the functionality of the proposed method. The proposed P2DCCA has many advantages over 2DCCA where the most significant advantage is its ability to extend to a mixture of P2DCCA model. It may also be possible to develop a probabilistic Bayesian model for P2DCCA and gaining the benefits of a Bayesian model. These are our future works.

References

1. I. Jolliffe, Principal component analysis, Wiley Online Library, 2002.
2. H. Hotelling, Relations between two sets of variates, Biometrika (1936) 321–377.
3. C.-C. Jia, S.-J. Wang, X.-J. Peng, W. Pang, C.-Y. Zhang, C.-G. Zhou, Z.-Z. Yu, Incremental multi-linear discriminant analysis using canonical correlations for action recognition, Neurocomputing 83 (2012) 56–63.
4. Y. R. Wang, K. Jiang, L. J. Feldman, P. J. Bickel, H. Huang, Inferring gene association networks using sparse canonical correlation analysis, arXiv preprint arXiv:1401.6504.
5. S. Huang, J. Chen, Z. Luo, Sparse tensor cca for color face recognition, Neural Computing and Applications 24 (7-8) (2014) 1647–1658.
6. J. Ye, R. Janardan, Q. Li, Gpca: an efficient dimension reduction scheme for image compression and retrieval, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2004, pp. 354–363.
7. D. Zhang, Z.-H. Zhou, (2d) 2pca: Two-directional two-dimensional pca for efficient face representation and recognition, Neurocomputing 69 (1) (2005) 224–231.
8. S. H. Lee, S. Choi, Two-dimensional canonical correlation analysis, Signal Processing Letters, IEEE 14 (10) (2007) 735–738.
9. M. E. Tipping, C. M. Bishop, Probabilistic principal component analysis, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 61 (3) (1999) 611–622.
10. B. Thompson, Exploratory and confirmatory factor analysis: Understanding concepts and applications., American Psychological Association, 2004.
11. M. W. Browne, The maximum-likelihood solution in inter-battery factor analysis, British Journal of Mathematical and Statistical Psychology 32 (1) (1979) 75–86.
12. A. Klami, S. Kaski, Probabilistic approach to detecting dependencies between data sets, Neurocomputing 72 (1) (2008) 39–46.
13. C. Archambeau, N. Delannay, M. Verleysen, Robust probabilistic projections, in: Proceedings of the 23rd International conference on machine learning, ACM, 2006, pp. 33–40.
14. C. Archambeau, F. R. Bach, Sparse probabilistic projections, in: Advances in neural information processing systems, 2009, pp. 73–80.
15. J. Zhao, P. L. Yu, J. T. Kwok, Bilinear probabilistic principal component analysis, Neural Networks and Learning Systems, IEEE Transactions on 23 (3) (2012) 492–503.
16. S. Ioffe, Probabilistic linear discriminant analysis, in: Computer Vision–ECCV 2006, Springer, 2006, pp. 531–542.
17. S. Kaski, J. Peltonen, Informative discriminant analysis, in: ICML, 2003, pp. 329–336.
18. F. R. Bach, M. I. Jordan, A probabilistic interpretation of canonical correlation analysis.
19. R. R. Sarvestani, R. Boostani, Ff-skpcca: Kernel probabilistic canonical correlation analysis, Applied Intelligence (2016) 1–17.
20. A. Podosinnikova, F. Bach, S. Lacoste-Julien, Beyond cca: Moment matching for multi-view models, in: Proc. 33rd International Conference on Machine Learning, 2016.
21. T. Michaeli, W. Wang, K. Livescu, Nonparametric canonical correlation analysis, in: http://arxiv.org/abs/1511.04839, 2016.

22. M. Safayani, M. T. M. Shalmani, Matrix-variate probabilistic model for canonical correlation analysis, EURASIP Journal on Advances in Signal Processing 2011 (1) (2011) 748430.
23. H. Wang, S. Chen, Z. Hu, B. Luo, Probabilistic two-dimensional principal component analysis and its mixture model for face recognition, Neural Computing and Applications 17 (5-6) (2008) 541–547.
24. D. Tao, M. Song, X. Li, J. Shen, J. Sun, X. Wu, C. Faloutsos, S. J. Maybank, Bayesian tensor approach for 3-d face modeling, Circuits and Systems for Video Technology, IEEE Transactions on 18 (10) (2008) 1397–1410.
25. A. Martinez, R. Benavente, The ar face database, CVC Technical Report 24.
26. D. B. Graham, N. M. Allinson, Characterising virtual eigensignatures for general purpose face recognition, in: Face Recognition, Springer, 1998, pp. 446–456.
27. J. Ye, R. Janardan, Q. Li, Two-dimensional linear discriminant analysis., in: NIPS, 2004, pp. 1569–1576.
28. `http://www.sheffield.ac.uk/eee/research/iel/research/face`.

A

As it is mentioned in Section 3, each column of the latent matrix Z has the distribution $N(0, I)$. Furthermore, based on (13) and (14) we can write:

$$p(\tau_{n,j}^l | z_{n,j}^l) \sim N(U z_{n,j}^l + m_j^l, \Psi^l) \tag{35}$$

$$p(\tau_{n,j}^r | z_{n,j}^r) \sim N(V z_{n,j}^r + m_j^r, \Psi^r). \tag{36}$$

Now from (20) and (27) and the distribution of columns of Z, we have

$$p(T_{1,n}^l, T_{2,n}^l, Z_n^l) = \prod_{j=1}^{n'} [(2\pi)^{-\frac{m_1+m_2}{2}} |\Psi^l|^{-\frac{1}{2}} \tag{37}$$

$$exp(-\frac{1}{2}(\tau_{n,j}^l - U z_{n,j}^l - m_j^l)^T (\Psi^l)^{-1} (\tau_{n,j}^l - U z_{n,j}^l - m_j^l))$$

$$(2\pi)^{-\frac{m'}{2}} exp(-\frac{1}{2}(z_{n,j}^l)^T (z_{n,j}^l))],$$

$$p(T_{1,n}^r, T_{2,n}^r, Z_n^r) = \prod_{j=1}^{m'} [(2\pi)^{-\frac{n_1+n_2}{2}} |\Psi^r|^{-\frac{1}{2}} \tag{38}$$

$$exp(-\frac{1}{2}(\tau_{n,j}^r - V z_{n,j}^r - m_j^r)^T (\Psi^r)^{-1} (\tau_{n,j}^r - V z_{n,j}^r - m_j^r))$$

$$(2\pi)^{-\frac{n'}{2}} exp(-\frac{1}{2}(z_{n,j}^r)^T (z_{n,j}^r))],$$

where $|A|$ denotes the determinant of matrix A.

In the E-step, expectation of log likelihood for each of the probabilistic models is calculated as:

$$E(L_c^l) = \sum_{n=1}^{N} \sum_{j=1}^{n'} [-\frac{m_1+m_2}{2} log(2\pi) - \frac{1}{2} log|\Psi^l| \tag{39}$$

$$-\frac{1}{2} tr\{(\Psi^l)^{-1} (\tau_{n,j}^l)(\tau_{n,j}^l)^T\} + tr\{(E[(z_{n,j}^l)] - m_j^l)^T U^T (\Psi^l)^{-1} \tau_{n,j}^l\}$$

$$-\frac{1}{2} tr\{(E[z_{n,j}^l] - m_j^l)^T U^T (\Psi^l)^{-1} U(E[z_{n,j}^l] - m_j^l)\} - \frac{m'}{2} log(2\pi)$$

$$-\frac{1}{2} tr\{E[(z_{n,j}^l)(z_{n,j}^l)^T]\}]$$

$$E(L_c^r) = \sum_{n=1}^{N} \sum_{j=1}^{m'} [-\frac{n_1+n_2}{2} log(2\pi) - \frac{1}{2} log|\Psi^r| \tag{40}$$

$$-\frac{1}{2} tr\{(\Psi^r)^{-1} (\tau_{n,j}^r)(\tau_{n,j}^r)^T\} + tr\{(E[(z_{n,j}^r)] - m_j^r)^T V^T (\Psi^r)^{-1} \tau_{n,j}^r\}$$

$$-\frac{1}{2} tr\{(E[z_{n,j}^r] - m_j^r)^T V^T (\Psi^r)^{-1} V(E[z_{n,j}^r] - m_j^r)\} - \frac{n'}{2} log(2\pi)$$

$$-\frac{1}{2} tr\{E[(z_{n,j}^r)(z_{n,j}^r)^T]\}]$$

$$E[z_{n,j}^l] = (M^l)^{-1} U^T (\Psi^l)^{-1} (\tau_{n,j}^l - m_j^l) \tag{41}$$

$$E[(z_{n,j}^l)(z_{n,j}^l)^T] = (M^l)^{-1} + E[(z_{n,j}^l)]E[(z_{n,j}^l)]^T \tag{42}$$

$$E[z_{n,j}^r] = (M^r)^{-1} V^T (\Psi^r)^{-1} (\tau_{n,j}^r - m_j^r) \tag{43}$$

$$E[(z_{n,j}^r)(z_{n,j}^r)^T] = (M^r)^{-1} + E[(z_{n,j}^r)]E[(z_{n,j}^r)]^T. \tag{44}$$

By obtaining formulas of $E(L_c^l)$ and $E(L_c^r)$, the M-step is done by derivation of each expected log-likelihoods.