A latent variable model for two-dimensional canonical correlation analysis and the variational inference

Mehran Safayani¹ · Saeid Momennzadeh¹ · Abdolreza Mirzaei¹ · Masoomeh Sadat Razavi¹

Published online: 6 April 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

The probabilistic dimension reduction has been and is a major concern. Probabilistic models provide a better interpretability of the dimension reduction methods and present a framework for their further extensions. In pattern recognition problems, data that have a matrix or tensor structure is initially transformed into a vector format. This eliminates the internal structure of the data. The available perspective is to maintain the internal structure of each data while reducing the dimensionality, which can reduce the small sample size problem. Canonical correlation analysis is one of the most important techniques in dimension reduction in multi-view data. A two-dimensional canonical correlation analysis as an extension of canonical correlation analysis has been proposed to preserve the matrix structure of the data. Here, a new probabilistic framework for two-dimensional canonical correlation analysis is proposed, where the matrix-variate distributions are applied to model the relation between the latent matrix and the two-view matrix-variate observed data. These distributions, specific to the matrix data, can provide better understanding of two-dimensional canonical correlation analysis and pave the way for further extensions. In general, there does not exist any analytical maximum likelihood solution for this model; therefore, here the two approaches, one based on the expectation maximization and other on variational expected maximization, are proposed for learning the model parameters. The synthetic data are applied to evaluate the convergence and quality of the mapping of these algorithms. The functionalities of these methods and their counterparts are compared on the real face datasets.

Keywords

Canonical correlation analysis · Probabilistic dimension reduction · Matrix-variate distribution · Variational expectation maximization

1 Introduction

Dimension reduction is one of the essential steps in the data mining process, which makes data analyses easier and faster for the learning algorithms. Canonical correlation analysis (CCA) is a well-known technique for the multi-view data dimension reduction. Bach and Jordan (2005) presented probabilistic interpretation of CCA and named it the probabilistic CCA (PCCA). Recently, these have become a major focus among researchers in this fields (Ju et al. 2018, 2016; Ju et al. 2015) and are very advantageous in handling of missing and outlier data (Chen et al. 2009), automatic selection of the appropriate projection vectors (Bishop 1999) and extension of the standard DR methods in to more complex cases like mixture and nonlinear models (Tipping and Bishop 1999; Zhao 2014; Lawrence 2005; Titsias and Lawrence 2010). Several extensions of PCCA like the Bayesian CCA (Klami et al. 2013), nonlinear Bayesian CCA (Damianou et al. 2016; Michaeli et al. 2016; Sarvestani and Boostani 2016) and variational Bayesian CCA (Wang 2007; Viinikanoja et al. 2010) are proposed. In these algorithms, it is assumed that the observed data are represented as vectors, while most real-world data are of matrix or tensor structure. In this case, they initially reshape the data into vectors by concatenating the columns or rows which would disturb the spatial internal structure of the data. Two-dimensional CCA (2DCCA)
(Lee and Choi 2007; Sun et al. 2010) is an extension of CCA through which the observed data matrix is processed without any matrix to vector transformation. This model has low computational cost vs. CCA and is robust vs. the small sample size problem. Presenting a probabilistic model for 2DCCA which needs more concentration is a challenging task than the same for CCA. Safayani et al. (2018) introduced the probabilistic 2DCCA (P2DCCA), where the two models for columns and rows of the image named left and right probabilistic models are defined. In this algorithm as to learning the parameters, it is assumed that the parameters of the right probabilistic model are fixed, first the observations are projected on the right, and next, the parameters of the left probabilistic model are estimated through the expectation maximization (EM) algorithm. In this context, the same procedure holds true for learning the parameters of the other side. These two procedures are repeated until their convergence. In this approach, it is assumed that the columns/rows of the observation matrix are independent and its probability distribution is calculated by multiplying the probability of the corresponding columns/rows to one another. Because there exist two models for the rows and the columns, a pair of data cannot be mapped to a single latent space, which makes the model non-generative. The distributions designed for matrix data are not applied here, thus reducing interpretability.

In this article, a new probabilistic model is proposed for 2DCCA, where unlike the P2DCCA, the matrix-variate distributions are applied in modeling the relations of the observed random matrices. Therefore, assumption of independence of the columns/rows of the image is not considered. Defining the two independent models for columns and rows of the image is eliminated here which makes this model more interpretable. Here, two random matrices are related through a latent variable with a matrix-variate normal distribution. This matrix-variate normal has the two \( \Sigma \) and \( \Phi \) covariance matrices for modeling the distribution, indicating that it is related to a multivariate normal distribution the covariance matrix of which is the kronecker product of \( \Sigma \) and \( \Phi \). This indicates that a structure for the covariance matrix is considered leading to reduction in number of the free parameters. Because there exists no closed-form solution for learning the parameters of this model, two new approaches are proposed in this article, named:

1. Unilateral matrix-variate CCA (UMVCCA) where it is assumed that the latent variables are only projected from one side, and the parameters are optimized through the EM algorithm
2. Variational matrix-variate CCA (VMVCCA), where the assumption of the first approach is not considered and the latent variables are projected from both sides, and the parameters are optimized through an algorithm based on the variational EM algorithm

The results of both of these proposed algorithms are first validated through the synthetic data and next through the real face datasets including “NIR-VIS 2.0” (Li et al. 2013) and the extended YaleB (Georghiades et al. 2001). The results are compared with the available algorithms like CCA, Regularized Kernel-CCA (R-KCCA) (Bilenko and Gallant 2015), Cluster-CCA (C-CCA) (Rasiwasia et al. 2014), 2DCCA (Lee and Choi 2007), PCCA (Bach and Jordan 2005) and P2DCCA (Safayani et al. 2018).

The main contributions of the article are briefed as follows:

- A new probabilistic model for 2DCCA, which can be extended to more complex mixture and nonlinear models
- Achieving more interpretability for the probabilistic models by applying matrix-variate distributions, designed for matrix data
- Presenting two new algorithms named UMVCCA and VMVCCA based on EM and variational EM for learning the parameters
- Presenting some innovative experiments for evaluating the projection matrices and the convergence of these algorithms through synthetic data

The rest of this article is organized as follows: the literature is reviewed in Sect. 2, the proposed models are introduced in Sect. 3, the experiments are run in Sect. 4, and finally, the article is concluded in Sect. 5.

## 2 Literature review

### 2.1 Matrix-variate normal distribution

Because in two-dimensional probabilistic models the random matrices rather than random vectors are of concern, it is convenient to apply matrix-variate distributions. In general, these distributions are the 2D generalization of multivariate distributions (Gupta and Nagar 1999). Matrix-variate normal distribution \( X \sim MN(M, \Sigma, \Phi) \) is defined as follows:

\[
\text{posterior distribution is } \exp \left[ \text{tr} \left( -\frac{1}{2} \Sigma^{-1}(X - M)\Phi^{-1}(X - M)' \right) \right] \frac{(2\pi)^\frac{1}{2}mn|\Sigma|^{\frac{1}{2}|\Phi|^{\frac{1}{2}}}n}{},
\]

where \( X \in \mathbb{R}^{m\times n} \) is a random matrix, \( M \in \mathbb{R}^{m\times n} \) is the mean matrix, \( \Sigma \in \mathbb{R}^{m\times m} > 0 \) and \( \Phi \in \mathbb{R}^{n\times n} > 0 \) are the column and row covariance matrices, respectively, and \( \text{tr}(A) \) is the matrix \( A \) trace. It is observed that if \( X \) follows \( X \sim MN(M, \Sigma, \Phi) \), then \( \text{vec}(X) \sim N(\text{vec}(M), \Sigma \otimes \Phi) \), where \( \otimes \) is the kronecker product (Gupta and Nagar 1999).
2.2 Canonical correlation analysis (CCA)

CCA is a method adopted in finding the relations between two multivariate sets of variables (Hotelling 1936). CCA is applied in mapping each set of data to a common subspace, where the correlation between the two datasets is maximized. By assuming that \( x^1 \in \mathbb{R}^{m_1} \) and \( x^2 \in \mathbb{R}^{m_2} \) are two random vectors, the CCA finds the two \( w^1 \) and \( w^2 \) linear mappings, where the following relation is maximized:

\[
\text{arg} \max_{w^1, w^2} \text{cov}(w^1 x^1, w^2 x^2),
\]

\[\text{s.t. } \text{var}(w^1 x^1) = 1, \quad \text{var}(w^2 x^2) = 1.\]  \tag{2}

The optimal \( w^1 \) and \( w^2 \) are obtained by solving the following eigen problems:

\[
C_{11}^{-1} C_{12} C_{22}^{-1} C_{21} w^1 = \lambda^2 w^1, \quad C_{22}^{-1} C_{21} C_{11}^{-1} C_{12} w^2 = \lambda^2 w^2,
\]

where \( C_{11} \) and \( C_{22} \) are the autocovariance matrices of random vectors \( x^1 \) and \( x^2 \), respectively, and \( C_{12} \) is the cross-covariance matrix and \( \lambda^2 \) is the largest eigenvalue equal to the square of canonical correlations.

2.3 Two-dimensional CCA (2DCCA)

2DCCA is an extension of CCA with direct performance on matrix data (Lee and Choi 2007). Let \( \{X^j_n \in \mathbb{R}^{m \times n} | j = 1, \ldots, N\} \) be the realizations of random matrix variable \( X^j_n \). Without loss of generality, it is assumed that the random variables are of zero mean. The 2DCCA is applied in obtaining the projection vectors \( l^j_{12} = \text{arg max} \text{cov}(L^j_{1} R^j_{2}, X^j_n) \) and \( r^j_{12} = \text{arg max} \text{cov}(L^j_{1} R^j_{2}, X^j_n) \), which maximize the following optimization problem:

\[
\text{arg} \max_{l^j_{12}, r^j_{12}} \left( l^j_{1} X^j_n r^j_{1} \right),
\]

\[\text{s.t. } \text{var}(l^j_{1} X^j_n r^j_{1}) = 1, \quad \text{var}(l^j_{2} X^j_n r^j_{2}) = 1.\]  \tag{5}

Because there exists no closed-form solution for this problem, in 2DCCA, it is assumed that \( r^j_{12} \) are fixed once and after some simplifications Eq. (5) converts to Eq. (6):

\[
\text{arg} \max_{l^j_{1}, l^j_{2}} l^j_{1} \Sigma^j_{12} l^j_{2},
\]

\[\text{s.t. } l^j_{1} \Sigma^j_{11} l^j_{1} = 1, \quad l^j_{2} \Sigma^j_{22} l^j_{2} = 1, \quad l^j_{1} \Sigma^j_{12} l^j_{2} = 1.\]  \tag{6}

where \( \Sigma^j_{ij} = \frac{1}{N} \sum_{n=1}^{N} X^j_n r^j_{i} X^j_n r^j_{j} \) and next the \( l^j_{12} \) are fixed, leading to Eq. (7):

\[
\text{arg} \max_{l^j_{1}, r^j_{2}} r^j_{1} \Sigma^j_{12} r^j_{2},
\]

\[\text{s.t. } r^j_{1} \Sigma^j_{11} r^j_{1} = 1, \quad r^j_{2} \Sigma^j_{22} r^j_{2} = 1.\]  \tag{7}

where \( \Sigma^j_{ij} = \frac{1}{N} \sum_{n=1}^{N} X^j_n r^j_{i} X^j_n r^j_{j} \). By solving Eqs. (6) and (7) iteratively, until their convergence, the optimum transformations \( \{l^j_{1}, r^j_{1}, l^j_{2}, r^j_{2}\} \) are yielded. It can be shown that Eqs. (6) and (7) are converted into the following eigen problems:

\[
\begin{bmatrix}
0 & \Sigma^j_{12} \\
\Sigma^j_{21} & 0
\end{bmatrix}
\begin{bmatrix}
l^j_{1} \\
l^j_{2}
\end{bmatrix} = \lambda
\begin{bmatrix}
\Sigma^j_{11} & 0 \\
0 & \Sigma^j_{22}
\end{bmatrix}
\begin{bmatrix}
r^j_{1} \\
r^j_{2}
\end{bmatrix},
\]

\[
\begin{bmatrix}
0 & \Sigma^j_{12} \\
\Sigma^j_{21} & 0
\end{bmatrix}
\begin{bmatrix}
l^j_{1} \\
l^j_{2}
\end{bmatrix} = \lambda
\begin{bmatrix}
\Sigma^j_{11} & 0 \\
0 & \Sigma^j_{22}
\end{bmatrix}
\begin{bmatrix}
r^j_{1} \\
r^j_{2}
\end{bmatrix}.\]  \tag{8}

The \( d_1 \), the largest eigenvectors of Eq. (8), generates the columns of matrix \( L^j_{12} \); similarly, \( d_2 \), the largest eigenvectors of Eq. (9) generates the columns of matrix \( R^j_{12} \).

2.4 Probabilistic CCA (PCCA)

The following latent variable model is defined through PCCA (Bach and Jordan 2005):

\[
x^j = W^j z + \mu^j + \epsilon^j, \quad j \in \{1, 2\}, \]  \tag{10}

where \( x^j \) is the observation random vector, \( z \) is the latent vector with a multivariate normal distribution of zero mean and identity covariance matrix, \( \epsilon^j \) is the residual noise vector which follows multivariate normal distribution with expectation of zero and \( \Psi^j \) covariance matrix and \( \mu^j \) is the mean vector of random vector \( x^j \). In this model, \( x^1 \) and \( x^2 \), given in the latent variable \( z \), are independent. The following distributions are yielded from Eq. (10):

\[
p(x^j | z) = \mathcal{N}(W^j z + \mu^j, \Psi^j), \]  \tag{11}

\[
p(x | z) = \mathcal{N}(W^j z + \mu, \Psi), \]  \tag{12}

\[
p(x) = \mathcal{N}(\mu, \Sigma), \]  \tag{13}

where \( x = [x^1, x^2]' \), \( W = [W^1, W^2]' \), \( \mu = [\mu^1, \mu^2]' \), \( \Psi = [\Psi^1, \Psi^2]' \) and \( \Sigma = \begin{bmatrix}
W^1 W^1' + \Psi^1 & W^1 W^2' \\
W^2 W^1' & W^2 W^2' + \Psi^2
\end{bmatrix}.\)

Let \( x^j_{n, 1} \) and \( x^j_{n, 2} \) is a set of observation vectors. The maximum log-likelihood estimate of parameters \( \theta = (W, \Psi) \) is obtained by maximizing Eq. (14):
\[ \mathcal{L}(\theta) = \frac{N}{2} \log \Sigma + \frac{1}{2} \sum_{n=1}^{N} tr \Sigma^{-1}(x_n - \mu)(x_n - \mu)^T, \]
\[ + \text{const}, \]  
\tag{14} \]
yielding:
\[ W_{ML}^j = \tilde{S}_{jj} U_d^j S^j, \quad j \in \{1, 2\}, \]  
\tag{15} \]
\[ \Psi_{ML}^j = \tilde{S}_{jj} - W_{ML}^j W_{ML}^j, \quad j \in \{1, 2\}, \]  
\tag{16} \]
where \( S^1 \) and \( S^2 \) are the arbitrary matrices in a sense that \( S^1 S^2 = C_d \) and \( C_d \) is the diagonal matrix of the first \( d \) canonical directions, and \( U_d^j \) consists of the first \( d \) canonical directions and \( \tilde{S}_{jj} \) is the sample covariance matrix of \( x^j \). An iterative algorithm based on EM is proposed by (Bach and Jordan 2005) to maximize Eq. (14), where \( z_n \) are considered as the latent variables in the optimization algorithm and the complete log-likelihood is expressed as follows:
\[ \mathcal{L}(\theta) = \sum_{n=1}^{N} \left[ \ln P(x_n|z_n) + \ln P(z_n) \right]. \]  
\tag{17} \]
By inserting Eq. (12) into Eq. (17) and some mathematics, the expected value of log-likelihood function with respect to the posterior distribution is obtained through Eq. (18):
\[ Q(\theta^{(i)}) = \sum_{n=1}^{N} \left\{ -\frac{1}{2} |\Psi| - \frac{1}{2} (x_n^\prime \Psi^{-1} x_n) - \frac{1}{2} (z_n z_n^\prime) \right\}, \]  
\tag{18} \]
where
\[ \langle z_n \rangle = MW' \Psi^{-1} x_n, \]  
\tag{19} \]
\[ \langle z_n z_n^\prime \rangle = M + \langle z_n \rangle \langle z_n \rangle^T, \]  
\tag{20} \]
\[ M = (W' \Psi W + I)^{-1}. \]  
\tag{21} \]
By maximizing Eq. (18) with respect to the parameters and inserting of Eqs. (19) and (20) into its solution, the following final update equations are obtained:
\[ W_{t+1} = \tilde{S} \Psi_t^{-1} W_t M_t' (M_t + M_t' \Psi_t^{-1} \tilde{S} \Psi_t^{-1} W_t M_t)^{-1}, \]  
\tag{22} \]
\[ \Psi_{t+1} = \tilde{S} - \tilde{S} \Psi_t^{-1} W_t M_t W_{t+1}. \]  
\tag{23} \]

### 3 Latent variable model for two-dimensional CCA

This newly proposed model extends PCCA for matrix data as follows:
\[ X^j = L^j Z R^j' + M^j + \Xi^j, \quad j \in \{1, 2\}, \]  
\tag{24} \]
where \( j \) is the index of observation variable, \( X^j \in \mathbb{R}^{m^j \times n^j} \) and \( \Xi^j \in \mathbb{R}^{m^j \times n^j} \) are the observed variables and residual noise matrix, respectively, \( Z \in \mathbb{R}^{d_1 \times d_2} \) is the latent matrix variable, \( M^j \in \mathbb{R}^{m^j \times n^j} \) is the means of corresponding observed variable (without generality loss, we assume that the observed variables are of zero mean) and \( L^j \in \mathbb{R}^{m^j \times d_1} \) and \( R^j \in \mathbb{R}^{n^j \times d_2} \) are the left and right projection matrices, respectively. The latent and noise matrices have the following distributions:
\[ p(Z) = MN(0, I, I), \]  
\tag{25} \]
\[ p(\Xi^j) = MN(0, \Psi_L^j, \Psi_R^j), \quad \Psi_L^j, \Psi_R^j \succeq 0, \quad j \in \{1, 2\}, \]  
\tag{26} \]
where \( \Psi_L^j \in \mathbb{R}^{m^j \times m^j} \) and \( \Psi_R^j \in \mathbb{R}^{n^j \times n^j} \) are the positive-semidefinite covariance matrices of the noise matrix variables.

From the linear model Eq. (24) and the noise distributions Eq. (26), the conditional distribution of observation variables given in the latent matrix is computed through Eq. (27):
\[ p(X^j|Z) = MN(L^j Z R^j, \Psi_L^j, \Psi_R^j), \quad j \in \{1, 2\}. \]  
\tag{27} \]
This model is an extension to the PCCA (Bach and Jordan 2005), with the difference that here the observed, latent and noise variables are represented as matrices instead of vectors. The vector form of Eq. (24) is computed through the following equations:
\[ \text{vec}(X^j) = W^j \text{vec}(Z) + \text{vec}(M^j) \]
\[ + \text{vec}(\Xi^j), \quad j \in \{1, 2\}, \]  
\tag{28} \]
\[ p(\text{vec}(Z)) = N(0, I), \]  
\tag{29} \]
\[ p(\text{vec}(\Xi^j)) = N(0, \Psi_L^j), \quad \Psi_L^j \succeq 0, \quad j \in \{1, 2\}, \]  
\tag{30} \]
where \( \text{vec}(\cdot) \) is an operator that vectorizes the input matrix by concatenating its columns, and \( W^j = (R^j \otimes L^j) \) and \( \Psi^j = \Psi_R^j \otimes \Psi_L^j \). As observed, the projection matrix \( W^j \) is the Kronecker product of the right and left projection matrices, and similarly, the noise covariance matrix \( \Psi^j \) can be decomposed into right and left noise covariance matrices. Consequently, the number of free parameters of Eq. (28) is less than those of Eq. (10).
The joint probabilistic distribution for a data pair and the corresponding subspace representation is expressed as Eq. (31):

$$P(X^1, X^2, Z) = P(X^1|Z)P(X^2|Z)P(Z)$$  \hspace{1cm} (31)

In PCCA method, the likelihood function of observed data $P(X^1, X^2)$ is obtained through integrating the latent variable. Here, due to applying matrix-variate distributions, in general, both the $P(X^1, X^2)$ and posteriori distribution $P(Z|X^1, X^2)$ do not follow a matrix-variate normal distribution. To overcome this drawback, here two approaches are proposed for learning the parameters: (1) a simplified model by assuming only one projection matrix left/right in a sense that the posterior distribution is derived based on a matrix-variate normal distribution, and a solution is provided based on the EM algorithm. This approach is named unilateral matrix-variate CCA model (UMVCCA), and (2) a general model with both left and right projections, where the posterior distribution $P(Z|X^1, X^2)$ is estimated by applying a parametric matrix-variate normal distribution $q(Z)$, and a lower bound of the log-likelihood is maximized through variational EM algorithm (Jordan et al. 1999). This approach is named variational matrix-variate CCA model (VMVCCA).

### 3.1 Unilateral matrix-variate CCA (UMVCCA)

Here, either the left or right mapping matrices is replaced with identity matrix in Eq. (24), where selection of either has no effect in the generality of the method, and here, the left mapping matrix is selected; thus,

$$X_j = Z R_j^I + Z_j, \quad j \in \{1, 2\},$$  \hspace{1cm} (32)

$$p(Z_j) = MN(0, I, \Psi^*_R),$$  \hspace{1cm} (33)

where $Z \in \mathbb{R}^{m_1 \times d_2}$. Assuming $m_1 = m_2 = m$, the above equations are combined into a factor analysis model as follows:

$$X = Z R' + Z,$$  \hspace{1cm} (34)

where $X = [X^1, X^2] \in \mathbb{R}^{m \times (n_1+n_2^2)}$, $R = [R_1', R_2'] \in \mathbb{R}^{d_1 \times (n_1+n_2^2)}$ and $Z = [Z_1, Z_2] \in \mathbb{R}^{m \times (n_1+n_2^2)}$. Through Eq. (34), the following distributions are obtained:

$$p(\epsilon) = MN(0, I, \begin{bmatrix} \Psi^*_R & 0 \\ 0 & \Psi^*_R \end{bmatrix}),$$  \hspace{1cm} (35)

$$p(X|Z) = MN(Z R', I, \Psi^*_R),$$  \hspace{1cm} (36)

$$P(Z|X) = MN(S R' \Psi^*_R^{-1} X', I, S),$$  \hspace{1cm} (37)

$$S = (R' \Psi^*_R^{-1} R + I)^{-1},$$  \hspace{1cm} (38)

where $\Psi^*_R = \begin{bmatrix} \psi^*_R & 0 \\ 0 & \psi^*_R \end{bmatrix}$.

For learning the parameters of UMVCCA, the well-known EM algorithm is applied here, where $X^1_{n1}$ and $X^2_{n2}$ are defined as $N$ pairs of training data, and $\{X_n, Z_n\}_{n=1}^N$ is the complete data; thus, the complete log-likelihood is obtained as follows:

$$\mathcal{L}(\theta) = \sum_{n=1}^N \ln \{P(X_n, Z_n)\} = \sum_{n=1}^N \ln \{P(X_n|Z_n)P(Z_n)\}.$$  \hspace{1cm} (39)

The final EM update equations are

$$R^* = \tilde{\Sigma} \Psi^*_R^{-1} RS [m S + SR' \Psi^*_R^{-1} \tilde{\Sigma} \Psi^*_R^{-1} R S]^{-1},$$  \hspace{1cm} (40)

$$\Psi^*_R = \frac{1}{m} \tilde{\Sigma} - \frac{2}{m} SR' \Psi^*_R^{-1} \tilde{\Sigma} + \frac{1}{m} SR' \Psi^*_R^{-1} \Psi^*_R^{-1} R S R',$$  \hspace{1cm} (41)

where $\tilde{\Sigma} = \frac{1}{N} \sum_{n=1}^N X'_n X_n$ is data scatter matrix and $N$ is the number of training data. The derivations are presented in “Appendix A”.

The steps of UMVCCA are expressed in Algorithm 1.

#### Algorithm 1 UMVCCA algorithm

**Input:** $X^1_{n1}$ and $X^2_{n2}$, initialization of $R'$ with random matrices and $\Psi^*_R$ with identity matrices, for $j=1,2$

1: repeat
2: Update $R$ and $\Psi_R$ using (40) and (41).
3: until Change of $\mathcal{L}$ is smaller than a threshold

**Output:** $R'$ and $\Psi^*_R$, for $j=1,2$

### 3.2 Variational matrix-variate CCA (VMVCCA)

Here, it is assumed that in Eq. (24) both the left and right projection matrices are non-identity. For maximizing the likelihood function, estimating the posterior of latent variable $Z_n$ given in the observed variables $X^1_n$ and $X^2_n$ is of essence. While, in general, there exists no matrix-variate equation for this posterior, consequently the variational EM algorithm (Jordan et al. 1999) is applied to maximize the lower bound of the likelihood function. In variational EM, a parameterized variational distribution, $q(Z_n)$, is chosen to estimate the posteriori distribution, $P(Z|X^1, X^2)$ followed by optimizing its parameters. Here, matrix-variate normal distribution is considered for $q(Z_n)$ defined as Eq.(42):

$$q(Z_n) \sim MN(C_n, O, S),$$  \hspace{1cm} (42)

where $C_n \in \mathbb{R}^{d_1 \times d_2}$ is the mean matrix and $O \in \mathbb{R}^{d_1 \times d_1}$ and $S \in \mathbb{R}^{d_2 \times d_2}$ are the column and row covariance matrices, respectively. The variational parameters are optimized to maximize the lower bound $\mathcal{L}(q)$ of the data log-likelihood function expressed as Eq.(43):
\[ L(q) = \sum_n \int \ln \left[ \frac{P(X_n^1, X_n^2, Z_n)}{q(Z_n)} \right] q(Z_n) dZ_n \]

where \( \mathbb{E}_q \) is the expected value with respect to the variational distribution. The optimization of the variational parameters \( \{C_n, O, S\} \) yields the following variational E-step update equations:

\[ O^* = \left[ \frac{1}{d_0} \sum_{j=1}^2 \text{tr}[R_j^i \Psi^{j-1}_R R_j^i S] L_j^i \Psi^{j-1}_L L_j^i + \frac{1}{d_2} \text{tr}[S] \times I \right]^{-1} \]

\[ (44) \]

\[ S^* = \left[ \frac{1}{d_1} \sum_{j=1}^2 \text{tr}[L_j^i \Psi^{j-1}_L L_j^i O] R_j^i \Psi^{j-1}_R R_j^i + \frac{1}{d_1} \text{tr}[O] \times I \right]^{-1} \]

\[ (45) \]

\[ vec(C_n) = \sum_{j=1}^2 \left[ R_j^i \Psi^{j-1}_R R_j^i \otimes L_j^i \Psi^{j-1}_L L_j^i \right] + I \]

\[ (46) \]

After estimating \( q(Z_n) \), in the M-step of the algorithm, the update equations of other model parameters are obtained as follows:

\[ \Psi^{j*}_L = \left[ \frac{1}{Nn_j} P_L^j + \frac{1}{n_j} \text{tr}[R_j^i \Psi^{j-1}_R R_j^i S] (L_j^i O L_j^i) \right]^{-1} \]

\[ (47) \]

\[ \Psi^{j*}_R = \left[ \frac{1}{Nm_j} P_R^j + \frac{1}{m_j} \text{tr}[L_j^i \Psi^{j-1}_L L_j^i O] (R_j^i S R_j^i) \right]^{-1} \]

\[ (48) \]

\[ L_j^* = \left[ -\sum_{n=1}^N X_n^j \Psi^{j-1}_R R_j^i C_n \right] \left[ -Nm_j [R_j^i \Psi^{j-1}_R R_j^i S] O \right. \]

\[ \left. - \sum_{n=1}^N C_n R_j^i \Psi^{j-1}_R R_j^i C_n \right]^{-1} \]

\[ (49) \]

\[ R_j^* = \left[ -\sum_{n=1}^N C_n^j L_j^i \Psi^{j-1}_L L_j^i \right] \left[ -Nm_j [L_j^i \Psi^{j-1}_L L_j^i O] S \right. \]

\[ \left. - \sum_{n=1}^N C_n^j L_j^i \Psi^{j-1}_L L_j^i C_n \right]^{-1} \]

\[ (50) \]

The derivations are presented in “Appendix B”. The steps of VMVCCA are presented in Algorithm 2.

### 3.3 Dimension reduction

Observation data matrices are projected into the low-dimensional space through \( \{L_j^i, R_j^i\}_{j=1}^2 \), while it is more natural to apply probabilistic projection. Here, similar to PPCA, the observed data are represented into the low-dimensional space by using the posterior distribution mean (i.e., \( E(Z|X^1, X^2) \)), where that for the VMVCCA is the mean of \( q(Z|X^1, X^2) \). This equation holds true whenever there are two observation matrices. In the case of one observation matrix (for instance, \( X^1 \)), \( E[q(Z|X^1, M^2)] \) is considered as the low-dimensional representation, where \( M^2 \) is the average of other observation matrix (i.e., \( X^2 \)) in the training data.

### 4 Experiments

#### 4.1 Synthetic data

##### 4.1.1 Convergence of algorithm

For generating the synthetic data, first the elements of the projection matrices \( L_j^i \in R_{32 \times 15}^{2 \times 2} \) and \( R_j^i \in R_{32 \times 15}^{2 \times 2} \) are substituted from the samples obtained from the uniform distribution within zero and one range; next, for each pair of observed data, the latent matrix \( Z \in R_{15 \times 15}^{2} \) and the residual matrices \( Z_j^i \in R_{32 \times 32}^{2 \times 2} \) are sampled from \( \mathcal{N}(0, I, I) \) and \( \mathcal{N}(0, 0.11) \), respectively, and the \( \{X_j^i \in R_{32 \times 32}^{2 \times 2} \} \) are generated by computing Eq. (24). Here, one thousand pairs of samples are generated through this procedure. The VMVCCA algorithm is run to compute the Frobenius norm of each projection matrix in different iterations. The difference of the computed norms in the successive iterations is diagramed in Fig. 1. As observed, the algorithm converges after several iterations.

##### 4.1.2 Analyzing the probabilistic subspace

Here, one more one thousand pairs of data \( \{X_j^i \in R_{32 \times 32}^{2 \times 2} \} \) are generated be Eq. (24), with the difference that here the projections \( L_j^i \in R_{32 \times 15}^{2 \times 2} \) and \( R_j^i \in R_{32 \times 15}^{2 \times 2} \) are vectors and come from the uniform distribution within zero and one range. For each data, \( Z \in R_{1 \times 1}^{I} \) is a scalar sampled from \( \mathcal{N}(0, 1) \) distribution and the residual matrices \( Z_j^i \in R_{32 \times 32}^{2 \times 2} \) are sampled from \( \mathcal{N}(0, 0.11, 0.11) \).
A latent variable model for two-dimensional canonical correlation analysis...

Fig. 1 Distance between two successive iteration mappings of VMVCCA

Fig. 2 Distance between estimated and true latent space in varying iterations of VMVCCA

The VMVCCA is run to obtain $C_{n|n=1}^{1000}$, which are the estimated mean of posterior distribution in each iteration of algorithm. Following this, the Euclidean distance between the true latent space $Z_{n|n=1}^{1000}$ and the corresponding $C_{n|n=1}^{1000}$ is computed and plotted for different iterations, Fig. 2. As observed, the error is reduced with a tendency towards zero. To assess the effect of the number of training data, this experiment is repeated with different training samples within 10 to 1000 range. The result indicates that the accuracy of estimation is improved with an increased number of training samples, Fig. 3.

This experiment is applicable only for $1 \times 1$ latent space, because the obtained latent space leads to the true latent space up to a rotation and in general there exists no solution for obtaining the rotation matrices. In this latent space, there exists no rotational matrix and the true and learned latent variables are equaled up to a scaling factor magnitude of which can be removed by normalizing both latent spaces. The latent variable sign cannot be removed by this procedure and for canceling the sign of scaling factor, the Euclidean distance between the true latent space with both positive and negative sign of the learned subspace is computed, and the minimum distance is selected.

4.1.3 Analyzing the learned projections of UMVCCA

Here, the difference in data matrix sampling is that only the right projection vectors $R_j$ are involved. UMVCCA is run to obtain the learned projection vectors, Fig. 4, where without considering the sign and scaling factor, the true and the learned projection vectors are similar.

4.2 NIR-VIS 2.0 face dataset

The performance of these proposed algorithms is evaluated on NIR-VIS 2.0 (Li et al. 2013). This dataset contains the visible and near-infrared face images. The data are collected through four different sessions where in each session, some visible and infrared images are taken from the subjects. There exist 740 unique person-session subjects involved in the dataset consisting of 710 persons in one session and 15 persons in two sessions. Each subject can have 1-22 VIS and/or 5-50 NIR face images, while not all subjects have both VIS and NIR images. This fact reduces the number of person-
session subjects from 740 to 728. Here, 728 VIS and 728 NIR images are selected as the train data and 4333 VIS images as the test data. The face images are grayed, resized to $32 \times 32$ pixels and cropped in a sense that the eyes of all images are in the same coordinates, Fig. 5.

4.2.1 Image reconstruction

Here, first, the pairs of visible and infrared images are projected onto the low-dimensional subspace through VMVCCA; next, it is sought to reconstruct the original images, Fig. 6, where as observed, the reconstructed images are similar to the original images. Here, the compressed representation is $15 \times 15$ pixels obtained through Eq. (46).

4.2.2 Convergence

The Euclidean distance of the consecutive projection matrices in VMVCCA learning is shown in Fig. 7, where as observed, after some iterations the distance becomes zero oriented. A plot of lower bound of log-likelihood with respect to different iterations is shown in Fig. 8, where convergence of the algorithm after some iterations is evident.

4.2.3 Face recognition

The model is first trained with some pairs of VIS/NIR images, and next, each train and test data are projected into the low-dimensional space, and then, the Euclidean distance between each projected test image and all the projected train images is computed and the label of the train image with the
least distance is selected as the label of the test image. The corresponding relations for representing images into the low-dimensional space for each method are tabulated in Table 1.

In addition to the Euclidean distance between the projected test image and train images, Eq. (51) is applied:

$$\text{label}(X^j) = \arg\max_n \mathbb{E}_q[\ln P(X^j|Z_n)], \quad n = 1 \ldots N, \quad j \in \{1, 2\},$$

(51)

where $X^j$ is a test image. The conditional probability of a test image given in each one of the training images is calculated, and the label corresponding to the most probable image is selected as the test data label. The expansion of formula (51) is presented in “Appendix C”. This approach is named probabilistic test (ptest). The error rate of different algorithms with different numbers of features in the low-dimensional space is compared in Table 2. For UMVCCA, the number of features is 32 $\times$ $d$, where $d$ is the dimension of reduced feature space allowing the selection of appropriate $d$ in a sense that the number of obtained features is close to the number of features in the corresponding column. As observed in this table, the number of features in the first and second column is 25 and 100, respectively. Therefore, for UMVCCA, the values of $d$ are considered to be 1 and 3, respectively.

Because CCA and PCCA methods inherently have the small sample size problem, first the data to the lower dimension of 727 (one less than the number of training data) are projected through PCA, and next, the corresponding algorithms are run. This fact prevents the projection of the data on to 30 $\times$ 30 features, and consequently, their places are dashed. In this table, the VMVCCA with ptest criteria outperforms its counterparts.

### 4.3 Extended YaleB

This database contains 64 near-frontal images subject to different illumination conditions of 38 subjects (Georghiades et al. 2001). Here, 30 and 27 illumination conditions of 38 subjects are selected for train and test set, respectively, totaling 1140 images for training and 1026 images for test, Fig. 9. For training of the models, the available illumination conditions of each person are partitioned into two separate sets randomly, generating two sets of 570 images (15 illumination conditions $\times$ 38 subjects) for training. After training, the test images are projected into the low-dimensional space to compute the nearest training image upon which the label is obtained. The error rate of these proposed algorithms and other competing methods are tabulated in Table 3, where for CCA and PCCA, the data are projected into 569 dimensions through PCA. As observed here, the VMVCCA outperforms its counterparts.

### 4.4 Computational complexities

Here, it is assumed that the input image has the $m \times m$ pixel size, and for simplicity $d$ (the number of features in the latent space) is considered as $m$. In the E-STEP of algorithm 2, to compute Eqs. (44) and (45), the computational order is $O(m^3)$ and for Eq. (46) (with the assumption that the inversion matrix algorithm is of order $O(m^3)$) is $O(tm^6 + tmN^3)$, where $N$ is the number of the images in the training set and $t$ is the number of iteration EM algorithm. In the M-STEP of algorithm 2, for Eqs. (47) and (48), the computational order is $O(tm^3)$ and for Eqs. (49) and (50), is $O(tNm^3)$. Therefore, the overall order of this algorithm is $O(tm^6 + tNm^3)$, while the orders of CCA, PCCA and 2DCCA are $O(m^6 + Nm^4)$, $O(tm^6 + Nm^3)$ and $O(rm^3 + rNm^3)$, respectively, where in the last-order equation, $r$ is the number of iteration for alternation between left and right models in 2DCCA. To reduce the computational cost, more efficient matrix inversion algorithm proposed by Williams (2012) with $O(m^4.753)$ order can be applied, where the complexity of this newly proposed algorithm becomes $O(tm^{4.746} + tNm^{2.373})$. 

---

**Fig. 8** Lower bound of log-likelihood in VMVCCA algorithm in NIRVIS 2.0 dataset

**Table 1** Different methods and their low-dimensional representation formula

| Method            | Subspace representation |
|-------------------|-------------------------|
| CCA               | $W^jX^j$                |
| PCCA (Bach and Jordan 2005) | $\mathbb{E}[z|x]$          |
| 2DCCA (Lee and Choi 2007) | $L^jX^jR^j$                |
| P2DCCA (Safayani et al. 2018) | $\mathbb{E}[Z|X]$            |
| UMVCCA            | $\mathbb{E}[q(Z)]$       |
| VMVCCA            | $\mathbb{E}[q(Z)]$       |
Table 2 Comparison of error rate of different CCA-based methods on NIR-VIS 2.0

| Methods        | Subspace dimensions | Best (%) |
|----------------|---------------------|----------|
|                | 5 x 5 (%)          | 10 x 10 (%) | 15 x 15 (%) | 20 x 20 (%) | 25 x 25 (%) | 30 x 30 (%) |
| PCA+CCA        | 89.2                | 51.4      | 31.1        | 21.2        | 16.3        | –           | 16.3        |
| PCA+PCCA       | 58.5                | 29.3      | 20.7        | 16.6        | 15.4        | –           | 15.4        |
| R-KCCA(Linear) | 53.2                | 20.0      | 18.0        | 17.6        | 17.6        | 17.6        | 17.6        |
| R-KCCA (RBF)   | 49.7                | 21.1      | 18.7        | 18.2        | 18.0        | 18.0        | 18.0        |
| C-CCA          | 79.1                | 44.4      | 35.3        | 27.0        | 22.9        | 19.6        | 19.6        |
| 2DCCA          | 27.2                | 20.2      | 19.2        | 18.8        | 19.1        | 19.3        | 19.3        |
| P2DCCA         | 35.6                | 18.7      | 11.3        | 9.2         | 11.6        | 9.2         | –           |
| UMVCCA         | 22.9                | 15.6      | 25.1        | 29.6        | 30          | 30          | 30          |
| VMVCCA         | 78.6                | 29.6      | 17.4        | 17.3        | 17.1        | 17.6        | 17.1        |
| VMVCCA(ptest)  | 40                  | 13.2      | 9.4         | 9.1         | 7.5         | 7.5         | 7.5         |

Fig. 9 Images of three persons under 12 different illumination conditions from extended Yale face database B

Table 3 Comparison of error rate of different CCA-based methods on extended Yale face database B

| Methods       | Subspace dimensions | Best (%) |
|---------------|---------------------|----------|
|               | 5 x 5 (%)          | 10 x 10 (%) | 15 x 15 (%) | 20 x 20 (%) | 25 x 25 (%) | 30 x 30 (%) |
| PCA+CCA       | 79.9                | 43.0      | 28.6        | 21.8        | –           | –           | 21.8        |
| PCA+PCCA      | 55.4                | 27.3      | 21.6        | 20.1        | –           | –           | 20.1        |
| R-KCCA(Linear)| 23.2                | 15.4      | 14.4        | 14.5        | 14.5        | 14.5        | 14.4        |
| R-KCCA (RBF)  | 31.2                | 23.9      | 22.9        | 23.2        | 23.5        | 23.9        | 22.9        |
| C-CCA         | 34.5                | 20.0      | 18.5        | 18.4        | 18.4        | 18.7        | 18.4        |
| 2DCCA         | 37.4                | 27.6      | 25.5        | 23.7        | 22.4        | 22          | 22          |
| P2DCCA        | 27.4                | 14.9      | 10.0        | 9.8         | 9.8         | 10.7        | 9.8         |
| UMVCCA        | 41.9                | 29.7      | 27.3        | 27.4        | 24.5        | 24.4        | 24.4        |
| VMVCCA        | 82                  | 34.6      | 21.7        | 17.3        | 11.1        | 7.9         | 7.9         |
| VMVCCA(ptest) | 70.7                | 32.7      | 24.3        | 15.4        | 10.3        | 7.1         | 7.1         |

5 Conclusion

A probabilistic model for CCA is proposed here, where it works by the matrix-variate data like image matrices. Two iterative approaches are presented for learning the parameter. In the first approach, named unilateral matrix-variate CCA (UMVCCA), the model is restricted by mapping the latent matrix from one side (row or column), and its learning method is based on expectation maximization. In the second approach, named the variational matrix-variate CCA (VMVCCA), the latent matrix is mapped from both sides and the posterior distribution is estimated through a variational matrix-variate distribution, where its parameters are optimized through variational expectation maximization. These proposed algorithms are evaluated by applying syntectic and real data. The results indicated that these algorithms converge after a few iteration. In comparison with other CCA-based algorithms, the proposed algorithms outperform their counterparts in terms of recognition accuracy. This model can be extended to more complex models such as mixture models, nonlinear models and Bayesian models.
Compliance with ethical standards

Conflict of interest Mehran Safayani declares that he has no conflict of interest. Saeid Momenzadeh declares that he has no conflict of interest. Abdolreza Mirzaei declares that he has no conflict of interest. Masoomeh Sadat Razavi declares that she has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

A Derivation of update equations for UMVCCA

The expectation of likelihood function with respect to the posterior distribution is computed as follows:

\[
L(\theta) = \sum_{n=1}^{N} E_q \left[ \ln p(X_n | Z_n) p(Z_n) \right] \quad \text{(52)}
\]

Both Eqs. (36) and (25) are inserted into Eq. (52) followed by removing the constant terms, and Eq. (53) is yielded:

\[
L(\theta) = \sum_{i=1}^{N} E_q \left[ \ln |\Psi_R|^{-\frac{1}{2}} m 
- \frac{1}{2} \text{tr}[(X_i - Z_i R')\Psi_R^{-1}(X_i - Z_i R')] \right]. \quad \text{(53)}
\]

The derivative of \( L \) with respect to \( \Psi_R \) is obtained as follows:

\[
\frac{\partial L}{\partial \Psi_R} = -\frac{Nm}{2} (\Psi_R^{-1}) 
+ \frac{1}{2} \Psi_R^{-1} E_q \left[ \sum_{i=1}^{N} [(X_i - Z_i R') (X_i - Z_i R')] \right] \Psi_R^{-1}. \quad \text{(54)}
\]

Setting Eq. (54) equals to zero and multiplying \( \Psi_R \) on its right side, and Eq. (55) is yielded:

\[
\Psi_R = \frac{1}{Nm} E_q \left[ \sum_{i=1}^{N} [(X_i - Z_i R') (X_i - Z_i R')] \right]. \quad \text{(55)}
\]

After some mathematics, Eq. (56) is obtained:

\[
\Psi_R = \frac{1}{Nm} \left[ \sum_{i=1}^{N} [X_i X_i' - 2E_q[Z_i]X_i + R E_q[Z_i'Z_i] R'] \right]. \quad \text{(56)}
\]

where

\[
E_q[Z_i] = X_i \Psi_R^{-1} R S \quad \text{(57)}
\]

\[
E_q[Z_i'Z_i] = mS + SR' \Psi^{-1} X_i' X_i \Psi^{-1} RS. \quad \text{(58)}
\]

Equation (41) in Sect. 3.1 is the result of inserting Eqs. (57) and (58) into Eq. (56).

In the similar manner, first the derivative of \( L \) with respect to \( R \) is set to zero, next \( \Psi_R \) is multiplied on its right side, and then, Eq. (59) is obtained:

\[
R = \left[ \sum_{i=1}^{N} X_i E_q[Z_i] \right] \times \left[ \sum_{i=1}^{N} E_q[Z_i'Z_i] \right]^{-1}. \quad \text{(59)}
\]

Equation (40) in Sect. 3.1 is yielded by inserting Eqs. (57) and (58) into Eq. (59).

B Derivation of update equations for VMVCCA

The lower bound for the likelihood function is:

\[
L = \sum_{n} E_q \left[ \ln \left( \frac{P(X_n | Z_n) P(Z_n) p(Z_n)}{q_Z(Z_n)} \right) \right]. \quad \text{(60)}
\]

By inserting Eqs. (25), (27) and (42) into Eq. (60), Eq. (61) is yielded:

\[
L = \sum_{j=1}^{2} \sum_{n} E_q \left[ \ln \mathcal{M}(L^j Z_n R^j \Psi_L^j, \Psi_R^j) \right]
+ \sum_{n} E_q \left[ \ln \mathcal{M}(0, I, I) \right] - \sum_{n} E_q \left[ \ln q(Z_n) \right], \quad \text{(61)}
\]

where

\[
E_q[Z_n] = C_n, \quad \text{(62)}
\]

\[
E_q[Z_n R^j \Psi_R R^j Z_n'] = O tr[R^j \Psi_R R^j S] + C_n R^j \Psi_R R^j C_n. \quad \text{(63)}
\]

The derivative of \( L \) with respect to \( O \) is obtained as follows:

\[
\frac{\partial L}{\partial O} = -\frac{N}{2} \sum_{j=1}^{2} \text{tr}[R^j \Psi_R^{-1} R^j S] L^j \Psi_R^{-1} L^j
- \frac{N}{2} \text{tr}[S] \times I + \frac{Nd_2}{2} O^{-1}. \quad \text{(64)}
\]

Be setting this derivative to zero, Eq. (44) in Sect. 3.2 is derived.
Similarly, the update equation for $S$ can be obtained, where the derivation is omitted.

To compute the update equation of $C_n$, Eq. (65) is applied:

\[
\frac{\partial L}{\partial C_n} = \sum_{j=1}^{2} -L^j \psi^{-1}_L L^j C_n R^j \psi^{-1}_R R^j \\
+ L^j \psi^{-1}_L X_j \psi^{-1}_R R^j - C_n = 0. 
\tag{65}
\]

Equation (46) in Sect. 3.2 is derived by first applying the vec operator on Eq. (65) and next solving the resulted linear equation.

For computing Eq. (47) in Sect. 3.2, the following equation is yielded:

\[
\text{Eq. (68) is yielded:}
\]

where Eq. (49) is derived from it. The derivations of similar to that of $L^j$.

\[
\text{C Expansion of equation (51)}
\]

By applying Eqs. (27) and (42), the following equation is obtained:

\[
\mathbb{E}_q \left[ \ln P(X^i | Z_n) \right] = \mathbb{E}_q \left[ \ln \left( (2\pi)^{-d} \frac{1}{2} |\psi^j_R|^{-1} \right) \right] \\
\mathbb{E}_q \left[ \ln |\psi^j_R|^{-1} tr \left( \frac{1}{2} \psi^j_R (X^i - L^j Z_n R^j)' \right) \right] \\
\mathbb{E}_q \left[ \ln |\psi^j_R|^{-1} (X^i - L^j Z_n R^j)' \right] 
\tag{70}
\]

with further simplification, Eq. (71) is obtained:

\[
\mathbb{E}_q \left[ \ln P(X^i | Z_n) \right] = \\
- \frac{n^j}{2} \ln |\psi^j_R| - \frac{m^j}{2} \ln |\psi^j_R| \\
- \frac{1}{2} \mathbb{E}_q \left[ tr \left( \psi^j_R (X^i - L^j Z_n R^j)' \right) \right] \\
- 2 \mathbb{E}_q \left[ tr \left( \psi^j_R (X^i - L^j Z_n R^j)' \right) \right] + tr \left( \psi^j_R (X^i - L^j Z_n R^j)' \right) 
\tag{71}
\]

where the terms affected by $\mathbb{E}_q$ are underlined. By inserting Eqs. (62) and (63) into Eq. (71) and some simplification, Eq. (72) is obtained:

\[
\mathbb{E}_q \left[ \ln P(X^i | Z_n) \right] = \\
- \frac{1}{2} \left[ tr \left( \psi^j_R (X^i - L^j Z_n R^j)' \right) \right] \\
+ \left[ tr \left( \psi^j_R (X^i - L^j Z_n R^j)' \right) \right] \\
- \frac{1}{2} \left[ tr \left( \psi^j_R (X^i - L^j Z_n R^j)' \right) \right] + \text{Const} 
\tag{72}
\]

References

Bach FR, Jordan MI (2005) A probabilistic interpretation of canonical correlation analysis. Technical report, Department of Statistics, University of California, Berkeley

Bilenko N, Gallant J (2015) Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging. Front Neuroinform. https://doi.org/10.3389/fninf.2016.00049

Bishop CM (1999) Bayesian pca. In: Proceedings of the 1998 conference on advances in neural information processing systems II. MIT Press, Cambridge, pp 382–388

Chen T, Martin E, Montague G (2009) Robust probabilistic pca with missing data and contribution analysis for outlier detection. Comput Stat Data Anal 53(10):3706–3716

Damianou A, Lawrence N, Ek C (2016) Multi-view learning as a nonparametric nonlinear inter-battery factor analysis. arXiv:1511.04839

Georghiades A, Belhumeur P, Kriegman D (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660

Gupta A, Nagar D (1999) Matrix variate distributions. PMS series. Addison-Wesley Longman, Limited, Reading
Hotelling H (1936) Relations between two sets of variates. Biometrika 28(3/4):321–377

Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. Mach Learn 37(2):183–233

Ju F, Sun Y, Gao J, Hu Y, Yin B (2015) Image outlier detection and feature extraction via l1-norm-based 2d probabilistic pca. IEEE Trans Image Process 24(12):4834–4846

Ju F, Sun Y, Gao J, Liu S, Hu Y, Yin B (2016) Mixture of bilateral-projection two-dimensional probabilistic principal component analysis. In: The IEEE conference on computer vision and pattern recognition (CVPR)

Ju F, Sun Y, Gao J, Hu Y, Yin B (2018) Vectorial dimension reduction for tensors based on bayesian inference. IEEE Trans Neural Netw Learn Syst 29(10):4579–4592

Klami A, Virtanen S, Kaski S (2013) Bayesian canonical correlation analysis. J Mach Learn Res 14(1):965–1003

Lawrence N (2005) Probabilistic non-linear principal component analysis with gaussian process latent variable models. J Mach Learn Res 6:1783–1816

Lee SH, Choi S (2007) Two-dimensional canonical correlation analysis. IEEE Signal Process Lett 14(10):735–738

Li S, Yi D, Lei Z, Liao S (2013) The casia nir-vis 2.0 face database.
In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 348–353

Michaeli T, Wang W, Livescu K (2016) Nonparametric canonical correlation analysis. arXiv:1511.04839

Rasiwasia N, Mahajan D, Mahadevan V, Aggarwal G (2014) Cluster canonical correlation analysis. In: Proceedings of the seventeenth international conference on artificial intelligence and statistics, vol 33, pp 823–831

Safayani M, Ahmadi SH, Afraabandpey H, Mirzaei A (2018) An EM based probabilistic two-dimensional cca with application to face recognition. Appl Intell 48(3):755–770

Sarvestani RR, Boostani R (2016) FF-SKPCCA: kernel probabilistic canonical correlation analysis. Appl Intell 46:438–454

Sun N, Ji Z, Cz Z, Zhao L (2010) Two-dimensional canonical correlation analysis and its application in small sample size face recognition. Neural Comput Appl 19(3):377–382

Tipping ME, Bishop CM (1999) Mixtures of probabilistic principal component analyzers. Neural Comput 11(2):443–482

Titsias M, Lawrence ND (2010) Bayesian Gaussian process latent variable model. In: Proceedings of the 13th international conference on artificial intelligence and statistics, Sardinia, Italy, Proceedings of machine learning research, vol 9, pp 844–851

Viinikanoja J, Klami A, Kaski S (2010) Variational Bayesian mixture of robust cca models. In: Proceedings of the 2010 European conference on machine learning and knowledge discovery in databases: Part III., ECML PKDD’10. Springer, Berlin, pp 370–385

Wang C (2007) Variational Bayesian approach to canonical correlation analysis. IEEE Trans Neural Netw 18(3):905–910. https://doi.org/10.1109/TNN.2007.891186

Williams VV (2012) Multiplying matrices faster than coppersmith-winograd. In: Proceedings of the forty-fourth annual ACM symposium on theory of computing, ACM, New York, NY, USA, STOC ’12, pp 887–898

Zhao J (2014) Efficient model selection for mixtures of probabilistic pca via hierarchical BIC. IEEE Trans Cybern 44(10):1871–1883

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.