Proceedings Paper:
Zhou, Y., Lu, H. and Cheung, Y.M. (2017) Bilinear Probabilistic Canonical Correlation Analysis via Hybrid Concatenations. In: 31st AAAI Conference on Artificial Intelligence, AAAI 2017. The 31st AAAI Conference on Artificial Intelligence (AAAI), 04-09 Feb 2017, San Francisco. Association for the Advancement of Artificial Intelligence, pp. 2949-2955. ISSN: 2159-5399. EISSN: 2374-3468.

Bilinear Probabilistic Canonical Correlation Analysis via Hybrid Concatenations

Yang Zhou[1], **Haiping Lu**[2], and **Yiu-ming Cheung**[1]
Department of Computer Science, Hong Kong Baptist University, Hong Kong, China[1]
Department of Computer Science, University of Sheffield, UK[2]
yangzhou@comp.hkbu.edu.hk, h.lu@sheffield.ac.uk, ymc@comp.hkbu.edu.hk

Abstract

Canonical Correlation Analysis (CCA) is a classical technique for two-view correlation analysis, while Probabilistic CCA (PCCA) provides a generative and more general viewpoint for this task. Recently, PCCA has been extended to bilinear cases for dealing with two-view matrices in order to preserve and exploit the matrix structures in PCCA. However, existing bilinear PCCAs impose restrictive model assumptions for matrix structure preservation, sacrificing generative correctness or model flexibility. To overcome these drawbacks, we propose BPCCA, a new bilinear extension of PCCA, by introducing a *hybrid* joint model. Our new model preserves matrix structures *indirectly* via hybrid vector-based and matrix-based concatenations. This enables BPCCA to gain more model flexibility in capturing two-view correlations and obtain close-form solutions in parameter estimation. Experimental results on two real-world applications demonstrate the superior performance of BPCCA over competing methods.

Introduction

Today's data are commonly collected from diverse sources or views that could represent different properties of the same object. For example, face images can be captured in different poses or illumination conditions, and webpage contents often include text, images, and hyperlinks. Canonical correlation analysis (CCA) (Hotelling 1936) is a classical method to learn the (linear) relationships between two sets of variables, i.e., data from two views. It seeks two transformations, one for each view, to project data into a common subspace in which the two views are maximally correlated. CCA has wide applications including information retrieval (Hardoon, Szedmak, and Shawe-Taylor 2004), multi-view clustering (Chaudhuri et al. 2009), and multi-label learning (Zhang and Schneider 2011).

CCA can also be viewed from a probabilistic perspective. Bach and Jordan (2005) gave a probabilistic interpretation of CCA, namely Probabilistic CCA (PCCA). Benefiting from the probabilistic framework, PCCA can capture data uncertainty, deal with missing values, and incorporate priori knowledge. Specifically, PCCA relates two-view observations $\boldsymbol{x}^{(1)} \in \mathbb{R}^{d_1}$ and $\boldsymbol{x}^{(2)} \in \mathbb{R}^{d_2}$ to a common latent variable $\boldsymbol{z} \in \mathbb{R}^q (1 \leqslant q \leqslant \min(d_1, d_2))$ as follows:

$$\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \boldsymbol{x}^{(v)} | \boldsymbol{z} \sim \mathcal{N}(\mathbf{W}^{(v)} \boldsymbol{z}, \mathbf{\Sigma}^{(v)}), \quad (1)$$

where we have assumed that data are centered with zero means, $v \in \{1, 2\}$ denotes the view, $\mathbf{W}^{(v)} \in \mathbb{R}^{d_v \times q}$ is the factor loading matrix, and $\mathbf{\Sigma}^{(v)} \in \mathbb{R}^{d_v \times d_v}$ is the covariance matrix.

The key idea of PCCA is to construct a joint model to capture the correlations of the two views. This is achieved via *vector-based concatenation* as follows:

$$\begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \end{bmatrix} | \boldsymbol{z} \sim \mathcal{N}(\begin{bmatrix} \mathbf{W}^{(1)} \\ \mathbf{W}^{(2)} \end{bmatrix} \boldsymbol{z}, \begin{bmatrix} \mathbf{\Sigma}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}^{(2)} \end{bmatrix}), \quad (2)$$

where $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ are concatenated as a joint observation generated by the common latent variable $\boldsymbol{z}$. With the same idea of two-view combination, several PCCA extensions have been proposed, including Bayesian CCA (Klami, Virtanen, and Kaski 2013) and nonlinear Bayesian CCA (Damianou, Lawrence, and Ek 2016).

CCA and PCCA are designed for vector inputs, while many real-world data are naturally in the form of tensors. Recently, some multilinear extensions of CCA have been proposed to learn correlations from tensors rather than vectors (Lee and Choi 2007; Yan et al. 2012; Gang et al. 2011; Lu 2013). These works show that exploiting the tensor structures in CCA could lead to compact subspace representation and robustness against the small sample size problem (Lu, Plataniotis, and Venetsanopoulos 2013).

In contrast, extending PCCA to its multilinear version is more challenging and has not been well-studied yet. One of the main challenges is how to construct a joint model for two-view combination while preserving the tensor structures. Of course, two-view tensors can be reshaped into vectors and then combined via *vector-based concatenation*. However, the resultant joint variable is a vector rather than a tensor whose structures have been lost in this way.

To preserve tensor structures in two-view combination, several bilinear PCCAs have been proposed. Two-dimensional Probabilistic CCA (2DPCCA) (Afrabandpey, Safayani, and Mirzaei 2014) takes partially projected (matrix) observations rather than the original ones as inputs, so that two views can be combined via *matrix-based concatenation* with preserved matrix (2D tensor) structures. However, this modification violates the generative nature of PCCA and fails to fully exploit the information from both views. Bayesian Multi-View Tensor Factorization (BMTF) (Khan and Kaski 2014) assumes that two-view matrix ob-

servations are generated by the same row (or column) factors so that they can be naturally combined in a bilinear model (via *matrix-based concatenation*) without vectorization. However, such restrictive assumption greatly limits the model flexibility and makes BMTF suitable only for two-view matrices with the *same row (or column) dimensions*.

In this paper, we propose a new bilinear PCCA named as Bilinear Probabilistic Canonical Correlation Analysis (BPCCA), which achieves both two-view combination and matrix structure preservation without violating the generative nature of PCCA or imposing restrictive model assumptions. Unlike existing works that seek only matrix-based concatenation of observations for *direct* structure preservation, we build BPCCA based on a *hybrid* joint model with *intermediate matrices*, where observations are combined via vector-based concatenation for more model flexibility while the intermediate matrices are combined via matrix-based concatenation for preserving matrix structures *indirectly*. Moreover, the hybrid model decouples the column and row structures from two-view matrices, which enables parameter estimation with close-form solutions.

Preliminaries and Related Works

Notations: Vectors and matrices are denoted by bold lowercase ($\boldsymbol{x}$) and uppercase ($\mathbf{X}$) letters, respectively. Symbol $\otimes$ denotes the Kronecker product. $\text{tr}(\cdot)$ is the matrix trace, $\text{vec}(\cdot)$ is the vectorization operator that stacks the columns of a matrix into a single column vector, $\text{diag}(\cdot)$ constructs a diagonal matrix from a vector, and $\text{blkdiag}(\cdot,\cdot)$ constructs a *block diagonal* matrix from two matrices. $\mathcal{N}_{d_c,d_r}(\mathbf{\Xi},\mathbf{\Sigma}_c,\mathbf{\Sigma}_r)$ denotes a *matrix-variate Gaussian distribution* with the mean matrix $\mathbf{\Xi}$, and the covariance matrices $\mathbf{\Sigma}_c \in \mathbb{R}^{d_c\times d_c}$ and $\mathbf{\Sigma}_r \in \mathbb{R}^{d_r\times d_r}$. If $\mathbf{X}\sim\mathcal{N}_{d_c,d_r}(\mathbf{\Xi},\mathbf{\Sigma}_c,\mathbf{\Sigma}_r)$, then $\text{vec}(\mathbf{X})\sim\mathcal{N}(\text{vec}(\mathbf{\Xi}),\mathbf{\Sigma}_r\otimes\mathbf{\Sigma}_c)$ (Gupta and Nagar 1999).

Given two-view observed matrices $\mathbf{X}^{(1)}\in\mathbb{R}^{d_1^c\times d_1^r}$ and $\mathbf{X}^{(2)}\in\mathbb{R}^{d_2^c\times d_2^r}$, our goal is to estimate a latent matrix $\mathbf{Z}\in\mathbb{R}^{q^c\times q^r}$ $(1\leqslant q^c\leqslant\min(d_1^c,d_2^c),1\leqslant q^r\leqslant\min(d_1^r,d_2^r))$ shared by the two views. It is common to relate the matrix observations of each view to the latent matrix via bilinear projections as follows (Xie et al. 2008; Yu, Bi, and Ye 2011):

$$\mathbf{X}^{(v)}=\mathbf{C}^{(v)}\mathbf{Z}{\mathbf{R}^{(v)}}^\top+\mathbf{E}^{(v)}, \tag{3}$$

where $v\in\{1,2\}$, $\mathbf{C}^{(v)}\in\mathbb{R}^{d_v^c\times q^c}$ and $\mathbf{R}^{(v)}\in\mathbb{R}^{d_v^r\times q^r}$ are the column and row factor matrices, respectively, and $\mathbf{E}^{(v)}$ is the noise matrix. For simplicity, we assume that data from the two views are centered with zero means.

Remarks: To estimate the common latent matrix $\mathbf{Z}$, we need to first combine the two views in a joint model. This can be easily done by *vector-based* concatenation, i.e., vectorizing both sides of (3) and concatenating the corresponding variables as (2), where $\boldsymbol{x}^{(v)}=\text{vec}(\mathbf{X}^{(v)})$, $\mathbf{W}^{(v)}=\mathbf{R}^{(v)}\otimes\mathbf{C}^{(v)}$, and $\boldsymbol{z}=\text{vec}(\mathbf{Z})$ due to the fact that $\text{vec}(\mathbf{C}^{(v)}\mathbf{Z}{\mathbf{R}^{(v)}}^\top)=(\mathbf{R}^{(v)}\otimes\mathbf{C}^{(v)})\text{vec}(\mathbf{Z})$. However, such vector-based concatenation breaks the matrix structure of $\mathbf{X}^{(v)}$, losing potentially useful structural information.

To address the above problem, 2DPCCA (Afrabandpey, Safayani, and Mirzaei 2014) takes partially projected observations $\mathbf{T}^{(v,c)}=\mathbf{X}^{(v)}\mathbf{R}^{(v)}\in\mathbb{R}^{d_v^c\times q^r}$ and $\mathbf{T}^{(v,r)}={\mathbf{X}^{(v)}}^\top\mathbf{C}^{(v)}\in\mathbb{R}^{d_v^r\times q^c}$ as inputs to avoid directly concatenating $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. This enables *matrix-based* concatenation of $\mathbf{T}^{(1,c)}$ and $\mathbf{T}^{(2,c)}$ as follows (a similar formulation hold for $\mathbf{T}^{(v,r)}$ with $\mathbf{R}^{(v)}$):

$$\begin{bmatrix}\mathbf{T}^{(1,c)}\\ \mathbf{T}^{(2,c)}\end{bmatrix}=\begin{bmatrix}\mathbf{C}^{(1)}\\ \mathbf{C}^{(2)}\end{bmatrix}\mathbf{Z}+\mathbf{E}_c, \tag{4}$$

where the joint observation maintains the column structures of the two-view observed matrices, and $\mathbf{E}_c$ is the column noise matrix. However, this model violates the *generative* nature of PCCA, since the true observations $\mathbf{X}^{(v)}$ can not be reconstructed from the latent matrix $\mathbf{Z}$ any more. In addition, the partial projections $\mathbf{T}^{(v,c)}$ and $\mathbf{T}^{(v,r)}$ only depend on $\mathbf{X}^{(v)}$, and fail to take the information from the other view into account.

Another recent method BMTF (Khan and Kaski 2014) assumes that two views share the same row (or column, equivalently) factor matrix $\tilde{\mathbf{R}}=\mathbf{R}^{(1)}=\mathbf{R}^{(2)}$, so that $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are naturally combined in the following bilinear model via *matrix-based* concatenation:

$$\begin{bmatrix}\mathbf{X}^{(1)}\\ \mathbf{X}^{(2)}\end{bmatrix}=\begin{bmatrix}\mathbf{C}^{(1)}\\ \mathbf{C}^{(2)}\end{bmatrix}\text{diag}(\boldsymbol{z})\tilde{\mathbf{R}}^\top+\begin{bmatrix}\mathbf{E}^{(1)}\\ \mathbf{E}^{(2)}\end{bmatrix}. \tag{5}$$

However, such restrictive assumption greatly limits the model flexibility and the applicability of BMTF. Since $\mathbf{R}^{(1)}=\mathbf{R}^{(2)}$, BMTF can only deal with two-view matrices with the same row (or column) dimensions.

Bilinear Probabilistic CCA

Both 2DPCCA and BMTF aim to preserve the matrix structures *directly* in the joint observation by avoiding vector-based concatenation as well as the subsequent vectorization. However, they achieve this at the expense of the generative nature or model flexibility. In this paper, we propose Bilinear Probabilistic CCA (BPCCA) with a *hybrid* joint model that allows both vector-based concatenation of observations to get rid of unnecessary model assumptions and matrix-based concatenation of intermediate matrices to preserve the matrix structures *indirectly*.

We present BPCCA in detail below. Firstly, we adapt a modified bilinear model for individual views, and then show how to combine the two views in a hybrid joint model with preserved matrix structures. Finally, we develop an EM-type algorithm for parameter estimation with close-form solutions.

Bilinear model for individual views: We characterize *individual* views with a modified bilinear model (Zhao, Yu, and Kwok 2012):

$$\begin{cases}\mathbf{X}^{(v)}=\mathbf{C}^{(v)}\mathbf{Z}{\mathbf{R}^{(v)}}^\top+\mathbf{F}_c^{(v)}+\mathbf{F}_r^{(v)}+\mathbf{E}^{(v)},\\ \mathbf{E}_c^{(v)}\sim\mathcal{N}_{d_v^c,q^r}(\mathbf{0},\mathbf{\Sigma}_c^{(v)},\mathbf{I}),\mathbf{E}_r^{(v)}\sim\mathcal{N}_{q^c,d_v^r}(\mathbf{0},\mathbf{I},\mathbf{\Sigma}_r^{(v)}),\\ \mathbf{E}^{(v)}\sim\mathcal{N}_{d_v^c,d_v^r}(\mathbf{0},\mathbf{\Sigma}_c^{(v)},\mathbf{\Sigma}_r^{(v)}),\mathbf{Z}\sim\mathcal{N}_{q^c,q^r}(\mathbf{0},\mathbf{I},\mathbf{I}),\end{cases}$$

where $\mathbf{F}_c^{(v)}=\mathbf{C}^{(v)}\mathbf{E}_r^{(v)}$, $\mathbf{F}_r^{(v)}=\mathbf{E}_c^{(v)}{\mathbf{R}^{(v)}}^\top$, and $\mathbf{E}_c^{(v)}$, $\mathbf{E}_r^{(v)}$, and $\mathbf{E}$ are column, row, and common noise matrices,

respectively. $\mathbf{\Sigma}_c^{(v)}$ and $\mathbf{\Sigma}_r^{(v)}$ are the column and row covariance matrices, respectively.

Originally, this model was proposed as a bilinear extension of Probabilistic *PCA* to learn subspaces for only one view. Here, we make use of it to model individual views in *CCA*. Compared with (3), two extra noise terms $\mathbf{C}^{(v)}\mathbf{E}_r^{(v)}$ and $\mathbf{E}_c^{(v)}\mathbf{R}^{(v)\top}$ are included. This improves the flexibility in capturing data uncertainty, and makes the marginal distribution $p(\mathbf{X}^{(v)})$ to be matrix-variable Gaussian, which is natural to model matrices.

The above model can also be rewritten as follows by decomposing the bilinear projection into *two stages* (Zhao, Yu, and Kwok 2012):

$$\begin{cases} \mathbf{X}^{(v)} & = \mathbf{Y}^{(v,c)}\mathbf{R}^{(v)\top} + \mathbf{F}^{(v,c)}, \\ \mathbf{Y}^{(v,c)} & = \mathbf{C}^{(v)}\mathbf{Z} + \mathbf{E}_c^{(v)}, \\ \mathbf{F}^{(v,c)} & = \mathbf{C}^{(v)}\mathbf{E}_r^{(v)} + \mathbf{E}^{(v)}, \end{cases} \tag{6}$$

where $\mathbf{Y}^{(v,c)} \in \mathbb{R}^{d_v^c \times q^r}$ is the column-projected intermediate matrix, and $\mathbf{F}^{(v,c)} \in \mathbb{R}^{d_v^c \times d_v^r}$ is the column-projected noise (residual) matrix. The conceptual meaning of this two-stage representation is that the latent matrix $\mathbf{Z}$ is first partially projected in the column direction onto $\mathbf{Y}^{(v,c)}$, and then $\mathbf{Y}^{(v,c)}$ is projected in the row direction to finally generate $\mathbf{X}^{(v)}$. Similarly, we can also decompose the bilinear projection by first projecting row and then column directions:

$$\begin{cases} \mathbf{X}^{(v)} & = \mathbf{C}^{(v)}\mathbf{Y}^{(v,r)} + \mathbf{F}^{(v,r)}, \\ \mathbf{Y}^{(v,r)} & = \mathbf{Z}\mathbf{R}^{(v)\top} + \mathbf{E}_r^{(v)}, \\ \mathbf{F}^{(v,r)} & = \mathbf{E}_c^{(v)}\mathbf{R}_x^{\top} + \mathbf{E}^{(v)}, \end{cases} \tag{7}$$

where $\mathbf{Y}^{(v,r)} \in \mathbb{R}^{q^c \times d_v^r}$ and $\mathbf{F}^{(v,r)} \in \mathbb{R}^{d_v^c \times d_v^r}$.

The three models above have the same marginal distribution $\mathbf{X}^{(v)} \sim \mathcal{N}_{d_v^c, d_v^r}(\mathbf{0}, \mathbf{\Psi}_c^{(v)}, \mathbf{\Psi}_r^{(v)})$, where $\mathbf{\Psi}_c^{(v)} = \mathbf{C}^{(v)}\mathbf{C}^{(v)\top} + \mathbf{\Sigma}_c^{(v)}$, and $\mathbf{\Psi}_r^{(v)} = \mathbf{R}^{(v)}\mathbf{R}^{(v)\top} + \mathbf{\Sigma}_r^{(v)}$. Therefore, they are equivalent.

With the above results, we tackle the problem of two-view combination with preserved matrix structures. Specifically, we combine the two views with preserved column and row structures based on (6) and (7), respectively. For compact presentation, we will provide detailed derivation only for combining the *column-wise model* (6), while the *row-wise model* (7) can be combined similarly.

Observed vector combination: We first combine the two-view observations. The conditional distribution of $\mathbf{X}^{(v)}$ given $\mathbf{Y}^{(v,c)}$ for individual views is $\mathbf{X}^{(v)}|\mathbf{Y}^{(v,c)} \sim \mathcal{N}_{d_v^c, d_v^r}(\mathbf{Y}^{(v,c)}\mathbf{R}^{(v)\top}, \mathbf{\Psi}_c^{(v)}, \mathbf{\Sigma}_r^{(v)})$. It can be rewritten in a vector form: $\boldsymbol{x}^{(v,c)}|\boldsymbol{y}^{(v,c)} \sim \mathcal{N}(\hat{\mathbf{R}}^{(v)}\boldsymbol{y}^{(v,c)}, \mathbf{\Psi}_c^{(v)} \otimes \mathbf{\Sigma}_r^{(v)})$, where $\boldsymbol{x}^{(v,c)} = \mathrm{vec}(\mathbf{X}^{(v)\top})$, $\boldsymbol{y}^{(v,c)} = \mathrm{vec}(\mathbf{Y}^{(v,c)\top})$, and $\hat{\mathbf{R}}^{(v)} = \mathbf{I} \otimes \mathbf{R}^{(v)}$. Since $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are independent given $\mathbf{Y}^{(1,c)}$ and $\mathbf{Y}^{(2,c)}$, the two-view observations $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ can be combined via vector-based concatenation as follows:

$$\boldsymbol{x}^c|\boldsymbol{y}^c \sim \mathcal{N}(\hat{\mathbf{R}}\boldsymbol{y}^c, \mathbf{L}_c), \tag{8}$$

where $\boldsymbol{x}^c = [\boldsymbol{x}^{(1,c)\top}, \boldsymbol{x}^{(2,c)\top}]^\top$, $\boldsymbol{y}^c = [\boldsymbol{y}^{(1,c)\top}, \boldsymbol{y}^{(2,c)\top}]^\top$, $\hat{\mathbf{R}} = \mathrm{blkdiag}(\hat{\mathbf{R}}^{(1)}, \hat{\mathbf{R}}^{(2)})$, and $\mathbf{L}_c = \mathrm{blkdiag}(\mathbf{\Psi}_c^{(1)} \otimes \mathbf{\Sigma}_r^{(1)}, \mathbf{\Psi}_c^{(2)} \otimes \mathbf{\Sigma}_r^{(2)})$. Compared with 2DPCCA and BMTF, we allow vector-based concatenation and do not require the joint observation $\boldsymbol{x}^c$ to be a matrix. As a result, the hybrid joint model is more flexible, since there is no need to impose additional assumptions on the observations.

Intermediate matrix combination: Although the matrix structures are not directly preserved in the joint observation $\boldsymbol{x}^c$ above, we can still preserve them *indirectly* in the intermediate matrices. Based on (6), the two-view intermediate matrices $\mathbf{Y}^{(1,c)}$ and $\mathbf{Y}^{(2,c)}$ can be directly combined via matrix-based concatenation as follows:

$$\mathbf{Y}^c|\mathbf{Z} \sim \mathcal{N}_{d_1^c + d_2^c, q^r}(\mathbf{C}\mathbf{Z}, \mathbf{\Sigma}_c, \mathbf{I}), \tag{9}$$

where $\mathbf{Y}^c = [\mathbf{Y}^{(1,c)\top}, \mathbf{Y}^{(2,c)\top}]^\top$, $\mathbf{C} = [\mathbf{C}^{(1)\top}, \mathbf{C}^{(2)\top}]^\top$, and $\mathbf{\Sigma}_c = \mathrm{blkdiag}(\mathbf{\Sigma}_c^{(1)}, \mathbf{\Sigma}_c^{(2)})$. Here, $\mathbf{Y}^c$ is generated by the joint factor matrix $\mathbf{C}$, and thus maintains the column structures of the two views. In other words, we preserve the matrix structures indirectly via the intermediate matrices.

With the above results, we can easily obtain other distributions involved in $\mathbf{Y}^c$ as follows:

$$\mathbf{Y}^c \sim \mathcal{N}_{d_1^c + d_2^c, q^r}(\mathbf{0}, \mathbf{\Psi}_c, \mathbf{I}), \tag{10}$$

$$\mathbf{Z}|\mathbf{Y}^c \sim \mathcal{N}_{q^c, q^r}(\mathbf{M}_c\mathbf{C}^\top\mathbf{\Sigma}_c^{-1}\mathbf{Y}^c, \mathbf{M}_c, \mathbf{I}), \tag{11}$$

where $\mathbf{\Psi}_c = \mathbf{C}\mathbf{C}^\top + \mathbf{\Sigma}_c$, and $\mathbf{M}_c = (\mathbf{C}^\top\mathbf{\Sigma}_c^{-1}\mathbf{C} + \mathbf{I})^{-1}$. Using the matrix-variate Gaussian property, it is easy to obtain the following distributions for $\boldsymbol{y}^c$:

$$\boldsymbol{y}^c \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi}_c \otimes \mathbf{I}), \tag{12}$$

$$\boldsymbol{y}^c|\boldsymbol{x}^c \sim \mathcal{N}(\mathbf{\Pi}_c\hat{\mathbf{R}}^\top\mathbf{L}_c^{-1}\boldsymbol{x}^c, \mathbf{\Pi}_c), \tag{13}$$

where $\mathbf{\Pi}_c = (\mathbf{\Psi}_c^{-1} \otimes \mathbf{I} + \hat{\mathbf{R}}^\top\mathbf{L}_c^{-1}\hat{\mathbf{R}})^{-1}$.

Remarks: The intermediate matrix $\mathbf{Y}^{(v,c)}$ can be viewed as a partial projection, which serves similar roles as $\mathbf{T}^{(v,c)}$ for matrix-based concatenation in 2DPCCA. However, 2DPCCA fails to relate $\mathbf{T}^{(v,c)}$ to $\mathbf{X}^{(v)}$ in a probabilistic model, and thus *breaks the generation path* from the latent matrix $\mathbf{Z}$ to the observation $\mathbf{X}^{(v)}$. Moreover, $\mathbf{T}^{(v,c)} = \mathbf{X}^{(v)}\mathbf{R}^{(v)}$ depends only on one view, while $\mathbf{Y}^{(1,c)}$ and $\mathbf{Y}^{(2,c)}$ jointly connect with both views (the joint observation $\boldsymbol{x}^c$) in $p(\boldsymbol{y}^c|\boldsymbol{x}^c)$ (13), and can be estimated more accurately.

Column-wise parameter estimation: The introduced hybrid joint model facilitates not only two-view combination but also parameter estimation. Provided a data set $\{\mathbf{X}_n^{(1)}, \mathbf{X}_n^{(2)}\}_{n=1}^N$ with N examples, our aim is to estimate BPCCA parameters $\boldsymbol{\theta}_c = \{\mathbf{C}, \mathbf{\Sigma}_c^{(1)}, \mathbf{\Sigma}_c^{(2)}\}$ and $\boldsymbol{\theta}_r = \{\mathbf{R}, \mathbf{\Sigma}_r^{(1)}, \mathbf{\Sigma}_r^{(2)}\}$. Unfortunately, it is difficult to solve $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}_r$ by maximizing the complete-data log-likelihood $\mathcal{L}(\boldsymbol{\theta}_c, \boldsymbol{\theta}_r) = \sum_{n=1}^N \ln p(\boldsymbol{x}_n^c|\boldsymbol{y}_n^c, \boldsymbol{\theta}_c, \boldsymbol{\theta}_r)p(\mathbf{Y}_n^c, \mathbf{Z}_n|\boldsymbol{\theta}_c) = \sum_{n=1}^N \ln p(\boldsymbol{x}_n^r|\boldsymbol{y}_n^r, \boldsymbol{\theta}_c, \boldsymbol{\theta}_r)p(\mathbf{Y}_n^r, \mathbf{Z}_n|\boldsymbol{\theta}_r)$, since $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}_r$ are coupled in $p(\boldsymbol{x}^c|\boldsymbol{y}^c)$ and $p(\boldsymbol{x}^r|\boldsymbol{y}^r)$.

To address this problem, we first consider to estimate $\boldsymbol{\theta}_c$ based on the column-wise two-stage model (6). If the intermediate matrices $\{\mathbf{Y}_n^c\}_{n=1}^N$ are observed, we can easily solve for $\boldsymbol{\theta}_c$ via the EM algorithm by maximizing $\mathcal{L}_c(\boldsymbol{\theta}_c) = \sum_{n=1}^N \ln p(\mathbf{Y}_n^c, \mathbf{Z}_n|\boldsymbol{\theta}_c) = \sum_{n=1}^N \ln p(\mathbf{Y}_n^c|\mathbf{Z}_n, \boldsymbol{\theta}_c)p(\mathbf{Z}_n)$

instead of the original log-likelihood $\mathcal{L}(\boldsymbol{\theta}_c, \boldsymbol{\theta}_r)$. This motivates us to obtain the statistics of $\{\mathbf{Y}_n^c\}_{n=1}^N$ via their maximum posteriori estimations according to $p(\boldsymbol{y}^c|\boldsymbol{x}^c)$ (13), and maximize the expectation of $\mathcal{L}_c(\boldsymbol{\theta}_c)$ w.r.t. $p(\boldsymbol{y}^c|\boldsymbol{x}^c)$ instead of the complicated $\mathcal{L}(\boldsymbol{\theta}_c, \boldsymbol{\theta}_r)$ for close-form solutions.

In the *E step*, we take the expectation of $\mathcal{L}_c(\boldsymbol{\theta}_c)$ w.r.t. $p(\mathbf{Z}, \mathbf{Y}^c|\boldsymbol{x}^c) = p(\mathbf{Z}|\mathbf{Y}^c)p(\boldsymbol{y}^c|\boldsymbol{x}^c)$ and obtain

$$\begin{aligned}\mathcal{Q}_c(\boldsymbol{\theta}_c) = -\frac{1}{2}\sum_{n=1}^{N}\Big\{q^r \ln|\boldsymbol{\Sigma}_c| + \mathrm{tr}(\langle\boldsymbol{\Sigma}_c^{-1}(\mathbf{Y}_n^c\mathbf{Y}_n^{c\top} \\ +\mathbf{C}\mathbf{Z}_n\mathbf{Z}_n^\top\mathbf{C}^\top - \mathbf{Y}_n^c\mathbf{Z}_n^\top\mathbf{C}^\top - \mathbf{C}\mathbf{Z}_n\mathbf{Y}_n^{c\top})\rangle_c)\Big\},\end{aligned} \tag{14}$$

where terms of $p(\mathbf{Z}_n)$ have been omitted as a constant, and $\langle\cdot\rangle_c$ denotes the expectation $\mathbb{E}[\mathbb{E}[\cdot|\mathbf{Y}^c]|\boldsymbol{x}^c]$ w.r.t. $p(\mathbf{Z}|\mathbf{Y}^c)$ and $p(\boldsymbol{y}^c|\boldsymbol{x}^c)$ correspondingly.

Define an operator $\mathrm{tr}_q(\mathbf{A})$ that generates an $m \times n$ matrix from an $mq \times nq$ block matrix $\mathbf{A}$ with $q \times q$ submatrices, where each element $\mathrm{tr}_q(\mathbf{A})_{ij}$ is the trace of the corresponding block $\mathbf{A}_{ij}$. With simple substitutions, we have $\mathbb{E}[\mathbf{Y}_n^c\mathbf{Y}_n^{c\top}|\boldsymbol{x}_n^c]$ of size $(d_1^c + d_2^c) \times (d_1^c + d_2^c)$ as the only statistic required for (14) as follows:

$$\mathbb{E}[\mathbf{Y}_n^c\mathbf{Y}_n^{c\top}|\boldsymbol{x}_n^c] = \mathrm{tr}_{q^r}(\mathbb{E}[\boldsymbol{y}_n^c\boldsymbol{y}_n^{c\top}|\boldsymbol{x}_n^c]), \tag{15}$$

where $\mathbb{E}[\boldsymbol{y}_n^c\boldsymbol{y}_n^{c\top}|\boldsymbol{x}_n^c] = \boldsymbol{\Pi}_c\hat{\mathbf{R}}^\top\mathbf{L}_c^{-1}\boldsymbol{x}_n^c\boldsymbol{x}_n^{c\top}\mathbf{L}_c^{-\top}\hat{\mathbf{R}}\boldsymbol{\Pi}_c + \boldsymbol{\Pi}_c$ from (13) is just a $(d_1^c + d_2^c)q^r \times (d_1^c + d_2^c)q^r$ block matrix with $q^r \times q^r$ submatrices (the covariance of the i-th and j-th rows of $\mathbf{Y}_n^c$ given $\boldsymbol{x}_n^c$).

In the *M step*, we maximize $\mathcal{Q}_c(\boldsymbol{\theta}_c)$ w.r.t. $\boldsymbol{\theta}_c$ and obtain:

$$\tilde{\mathbf{C}} = \left[\sum_{n=1}^{N}\langle\mathbf{Y}_n^c\mathbf{Z}_n^\top\rangle_c\right]\left[\sum_{n=1}^{N}\langle\mathbf{Z}_n\mathbf{Z}_n{}^\top\rangle_c\right]^{-1}, \tag{16}$$

$$\tilde{\boldsymbol{\Sigma}}_c^{(v)} = \frac{1}{Nq^r}\sum_{n=1}^{N}\langle\mathbf{H}_n^{(v,c)}\mathbf{H}_n^{(v,c)\top}\rangle_c, \tag{17}$$

where $\langle\mathbf{Y}_n^c\mathbf{Z}_n^\top\rangle_c = \mathbb{E}[\mathbf{Y}_n^c\mathbf{Y}_n^{c\top}|\boldsymbol{x}_n^c]\boldsymbol{\Sigma}_c^{-1}\mathbf{C}\mathbf{M}_c$, $\langle\mathbf{Z}_n\mathbf{Z}_n^\top\rangle_c = q^r\mathbf{M}_c + \mathbf{M}_c\mathbf{C}^\top\boldsymbol{\Sigma}_c^{-1}\langle\mathbf{Y}_n^c\mathbf{Z}_n^\top\rangle_c$, and $\mathbf{H}_n^{(v,c)} = \mathbf{Y}_n^{(v,c)} - \mathbf{C}^{(v)}\mathbf{Z}_n$.

Row-wise parameter estimation: With the hybrid concatenations in (8) and (9), we can combine the row-wise model (7) and estimate the parameter $\boldsymbol{\theta}_r$ similarly. Define $\boldsymbol{x}^{(v,r)} = \mathrm{vec}(\mathbf{X}^{(v)})$, $\boldsymbol{y}^{(v,r)} = \mathrm{vec}(\mathbf{Y}^{(v,r)})$, $\boldsymbol{x}^r = [\boldsymbol{x}^{(1,r)\top}, \boldsymbol{x}^{(2,r)\top}]^\top$, $\boldsymbol{y}^r = [\boldsymbol{y}^{(1,r)\top}, \boldsymbol{y}^{(2,r)\top}]^\top$, and $\mathbf{Y}^r = [\mathbf{Y}^{(1,r)}, \mathbf{Y}^{(2,r)}]^\top$. After obtaining the posterior distribution $p(\boldsymbol{y}^r|\boldsymbol{x}^r)$ like (13), $\boldsymbol{\theta}_r$ can be solved by maximizing the expectation of $\mathcal{L}_r(\boldsymbol{\theta}_r) = \sum_{n=1}^N \ln p(\mathbf{Y}_n^r, \mathbf{Z}_n|\boldsymbol{\theta}_r)$ w.r.t. $p(\boldsymbol{y}^r|\boldsymbol{x}^r)$ via the EM algorithm (see the supplementary material[1] for more details).

In the *E step*, we take the expectation of $\mathcal{L}_r(\boldsymbol{\theta}_r)$ w.r.t. $p(\mathbf{Z}, \mathbf{Y}^r|\boldsymbol{x}^r) = p(\mathbf{Z}|\mathbf{Y}^r)p(\boldsymbol{y}^r|\boldsymbol{x}^r)$ and compute the following expectation:

$$\mathbb{E}[\mathbf{Y}_n^r\mathbf{Y}_n^{r\top}|\boldsymbol{x}_n^r] = \mathrm{tr}_{q^c}(\mathbb{E}[\boldsymbol{y}_n^r\boldsymbol{y}_n^{r\top}|\boldsymbol{x}_n^r]), \tag{18}$$

[1]Available at https://drive.google.com/open?id=0B6F5rwPNzSmOZWM5M2FBWUp3OUk.

Algorithm 1 Bilinear Probabilistic CCA

Input: Data set $\{\mathbf{X}_n^{(1)}, \mathbf{X}_n^{(2)}\}_{n=1}^N$, regularization parameter γ, and initialized $\mathbf{C}^{(v)}, \mathbf{R}^{(v)}, \boldsymbol{\Sigma}_c^{(v)}, \boldsymbol{\Sigma}_r^{(v)}, v \in \{1, 2\}$.
Center the data and compute the regularized sample covariance.
repeat
  Compute the expectation $\mathbb{E}[\mathbf{Y}_n^c\mathbf{Y}_n^{c\top}|\boldsymbol{x}_n^c]$ via (15).
  Update $\mathbf{C}$ and $\boldsymbol{\Sigma}_c^{(v)}$ via (16) and (17), respectively.
  Compute the expectation $\mathbb{E}[\mathbf{Y}_n^r\mathbf{Y}_n^{r\top}|\boldsymbol{x}_n^r]$ via (18).
  Update $\mathbf{R}$ and $\boldsymbol{\Sigma}_r^{(v)}$ via (19) and (20), respectively.
until convergence.
Output: $\mathbf{C}^{(v)}, \mathbf{R}^{(v)}, \boldsymbol{\Sigma}_c^{(v)}, \boldsymbol{\Sigma}_r^{(v)}, v \in \{1, 2\}$.

where $\mathbb{E}[\boldsymbol{y}_n^r\boldsymbol{y}_n^{r\top}|\boldsymbol{x}_n^r]$ is a $(d_1^r + d_2^r)q^c \times (d_1^r + d_2^r)q^c$ block matrix with $q^c \times q^c$ submatrices and can be computed from $p(\boldsymbol{y}^r|\boldsymbol{x}^r)$. Let $\langle\cdot\rangle_r$ denote the expectation $\mathbb{E}[\mathbb{E}[\cdot|\mathbf{Y}^r]|\boldsymbol{x}^r]$ w.r.t. $p(\mathbf{Z}|\mathbf{Y}^r)$ and $p(\boldsymbol{y}^r|\boldsymbol{x}^r)$ correspondingly. In the *M step*, we maximize $\mathcal{Q}_r(\boldsymbol{\theta}_r) = \sum_{n=1}^N \ln\langle p(\mathbf{Y}_n^r, \mathbf{Z}_n|\boldsymbol{\theta}_r)\rangle_r$ w.r.t. $\boldsymbol{\theta}_r$, which leads to the following solutions:

$$\tilde{\mathbf{R}} = \left[\sum_{n=1}^{N}\langle\mathbf{Y}_n^r\mathbf{Z}_n\rangle_r\right]\left[\sum_{n=1}^{N}\langle\mathbf{Z}_n{}^\top\mathbf{Z}_n\rangle_r\right]^{-1}, \tag{19}$$

$$\tilde{\boldsymbol{\Sigma}}_r^{(v)} = \frac{1}{Nq^c}\sum_{n=1}^{N}\langle\mathbf{H}_n^{(v,r)}\mathbf{H}_n^{(v,r)\top}\rangle_r. \tag{20}$$

where $\langle\mathbf{Y}_n^r\mathbf{Z}_n\rangle_r = \mathbb{E}[\mathbf{Y}_n^r\mathbf{Y}_n^{r\top}|\boldsymbol{x}_n^r]\boldsymbol{\Sigma}_r^{-1}\mathbf{R}\mathbf{M}_r$, $\langle\mathbf{Z}_n^\top\mathbf{Z}_n\rangle_r = \mathbf{M}_r\mathbf{R}^\top\boldsymbol{\Sigma}_r^{-1}\langle\mathbf{Y}_n^r\mathbf{Z}_n^\top\rangle_r + q^c\mathbf{M}_r$, and $\mathbf{H}_n^{(v,r)} = \mathbf{Y}_n^{(v,r)} - \mathbf{R}^{(v)}\mathbf{Z}_n^\top$.

The BPCCA algorithm: By alternatively updating $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}_r$, we obtain the BPCCA algorithm. While a theoretical convergence guarantee is not yet available, we perform some empirical convergence studies and find that BPCCA is stable and has relatively fast convergence. Algorithm 1 gives the pseudocode of BPCCA.

Covariance regularization: For better generalization, it is common to regularize the sample covariance in CCA, which is also known as regularized CCA (Vinod 1976; Leurgans, Moyeed, and Silverman 1993). We incorporate such regularization into BPCCA as well. Notice that the sufficient statistics $\sum_{n=1}^N \mathbb{E}[\mathbf{Y}_n^c\mathbf{Y}_n^{c\top}|\boldsymbol{x}_n^c]$ and $\sum_{n=1}^N \mathbb{E}[\mathbf{Y}_n^r\mathbf{Y}_n^{r\top}|\boldsymbol{x}_n^r]$ are determined by the sample covariance matrices $\mathbf{S}_c = \frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n^c\boldsymbol{x}_n^{c\top}$ and $\mathbf{S}_r = \frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n^r\boldsymbol{x}_n^{r\top}$, respectively. In practice, $\mathbf{S}_c$ and $\mathbf{S}_c$ could be ill-conditioned, leading to unstable results. To solve this problem, we replace $\mathbf{S}_c$ and $\mathbf{S}_r$ with $\tilde{\mathbf{S}}_c = \mathbf{S}_c + \gamma\mathbf{I}$ and $\tilde{\mathbf{S}}_r = \mathbf{S}_r + \gamma\mathbf{I}$, respectively, where γ is a regularization parameter.

Initialization and prediction: We initialize the BPCCA parameters $\mathbf{C}^{(v)}$, $\mathbf{R}^{(v)}$, $\boldsymbol{\Sigma}_c^{(v)}$, $\boldsymbol{\Sigma}_r^{(v)}$ to identity matrices with proper sizes. After solving $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}_r$, we can project an observation $\mathbf{X}^{(v)}$ into the common subspace as follows:

$$\mathbb{E}[\mathbf{Z}|\mathbf{X}^{(v)}] = \mathbf{M}_c^{(v)}\mathbf{C}^{(v)\top}\boldsymbol{\Sigma}_c^{(v)-1}\mathbf{X}^{(v)}\boldsymbol{\Sigma}_r^{(v)-1}\mathbf{R}^{(v)}\mathbf{M}_r^{(v)},$$

where $\mathbf{M}_c^{(v)} = (\mathbf{C}^{(v)\top}\boldsymbol{\Sigma}_c^{(v)-1}\mathbf{C}^{(v)} + \mathbf{I})^{-1}$, and $\mathbf{M}_r^{(v)} = (\mathbf{R}^{(v)\top}\boldsymbol{\Sigma}_r^{(v)-1}\mathbf{R}^{(v)} + \mathbf{I})^{-1}$.

Time and space complexity: BPCCA has comparable time and space complexity as that of EM-based PCCA (Bach and Jordan 2005), which is dominated by the computations in (15) and (18). The straightforward implementation of BPCCA needs $O(Nd^4)$ and $O(d^4)$ for computing and storing the covariance matrix, respectively, $O(KNd^4)$ for matrix multiplication, and $O(K(dq)^3)$ for matrix inverse, where K is the number of iterations, and we have assumed that $d_1^c = d_2^c = d_1^r = d_2^r = d$, and $q^c = q^r = q$ for simplicity. Such time complexity could be further reduced by using the properties of matrix structures and Kronecker product.

Discussion on multi-view BPCCA: Although BPCCA is designed only for dealing with two views, the proposed hybrid concatenation strategy can be readily used to combine multi-view matrices for correlation learning. However, to develop a practical multi-view extension of BPCCA, we need to capture both common and view-specific components from multiple views. This could be achieved by extending inter-battery factor analysis (Browne 1979) to bilinear cases.

Experiments

This section evaluates BPCCA on two real-world applications: facial image matching and face photo-sketch recognition.

Algorithms and their settings: BPCCA is compared against ***linear baselines***: CCA, PCCA (Bach and Jordan 2005); ***bilinear CCAs:*** MCCA1+2 (Lu 2013), 2DCCA (Lee and Choi 2007); and ***bilinear PCCA:*** BMTF (Khan and Kaski 2014). Originally BMTF (Khan and Kaski 2014) is a Bayesian nonparametric method for tensor factorization rather than canonical correlation analysis, and is solved by time consuming sampling techniques. For simplicity and fair comparison, we implemented a non-Bayesian BMTF by estimating the parameters of (5) via a generalized EM algorithm (Meng and Rubin 1993). As found in preliminary studies, the original BMTF is too slow to extract hundreds of features and achieves much worse performance than its non-Bayesian version. We also implemented 2DPCCA, and found it numerical unstable with poor results, which is likely due to its broken probabilistic framework.

For $a \times b$ matrices, we extract the maximum number of features for CCA, 2DCCA, and MCCA1+2, i.e., ab, ab, and $\min(a, b)$ features, respectively. For PCCA, BPCCA, and BMTF, we set the number of extracted features to $ab - 1$, $(a-1)(b-1)$, and $ab-1$, respectively. The extracted features are then sorted by the corresponding correlation coefficient ρ in descending order, where ρs computed by both training and test examples are tested. For CCA, 2DCCA, and MCCA1+2, a heuristic scheme that further weights each extracted feature z_i by the corresponding ρ_i is also tested (Weenink 2003), which usually leads to better results for these non-probabilistic CCAs. We test 2DCCA and MCCA1+2 with up to 10 iterations, after verifying that more iterations do not result in statistically significant improvement in accuracy. For probabilistic methods, we iterate PCCA and BPCCA until the log-likelihood converges (a relative change is smaller than 10^{-5}), or up to 500 iterations.

The nearest neighbor classifier is used for matching, where we test L_1, L_2, and $\cos$ metrics to measure the distances. Covariance matrix regularization as in BPCCA is also applicable to all the competing methods. We employ this strategy for all the methods and show their best results, where the regularization parameter is selected from $\gamma \in \{0, 10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. We highlight the best and comparable results in **bold font** based on t-test with a p-value of 0.055 and underline the second best ones.

Table 1: Average rank-one matching accuracy on the PIE data set (**Best**; Second best).

Pose	0° vs. 22.5°	0° vs. -22.5°	22.5° vs. -22.5°
CCA	90.86 ± 4.04	91.28 ± 5.52	74.25 ± 6.02
2DCCA	82.22 ± 5.99	77.70 ± 9.52	59.52 ± 13.44
MCCA1+2	87.28 ± 6.37	88.04 ± 6.75	68.65 ± 8.12
PCCA	93.12 ± 3.75	90.93 ± 5.17	$\mathbf{75.53 \pm 8.81}$
BMTF	90.11 ± 3.41	90.70 ± 4.28	70.54 ± 5.06
BPCCA	$\mathbf{94.50 \pm 3.97}$	$\mathbf{94.74 \pm 3.48}$	$\mathbf{80.58 \pm 5.03}$

Matching facial images of different poses: A subset of the PIE database (Sim, Baker, and Bsat 2003) is tested. Face images in three poses (C27, C29, C05) corresponding to yaw angle of 0°, 22.5°, -22.5° and 18 illumination conditions (02$\sim$06, 10$\sim$22) are selected to form 3672 faces from 68 subjects. Due to missed faces, illuminations 07$\sim$09 are excluded. Each image is cropped and normalized to 32×32 graylevel pixels.

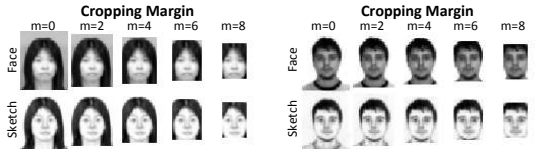
In this study, face images of each pose are considered as a view and treated as the probe set, while images from another view serve as the gallery set. Our aim is to match the probe and gallery images after projecting them into a common subspace, where a correct match means that the probe and gallery are of *both the same subject and illumination condition*. We show the matching performance on 3 paired poses including 0° vs. 22.5°, 0° vs. -22.5°, and 22.5° vs. -22.5°, where the training and test sets are partitioned by 10-fold CV w.r.t. subjects. More results for 22.5° vs. 0°, -22.5° vs. 0°, and -22.5° vs. 22.5° can be found in the supplementary material.

As seen in Table 1, PCCA obtains the second best performance, which indicates the advantages of probabilistic CCAs. BPCCA consistently outperforms other methods and achieves statistically significant improvements in most cases. This shows the advantages of hybrid concatenations in gaining more model flexibility and preserving matrix structures. On the other hand, BMTF is inferior to even baselines CCA and PCCA, which can be attributed to its limited model flexibility.

Face photo-sketch recognition: In this experiment, the CUHK face-sketch database (CUFS) (Tang and Wang 2003) is tested. It consists of the Chinese University of Hong Kong (CUHK) student database (Tang and Wang 2003), the AR databases (Martinez 1998), and the XM2VTS database (Messer et al. 1999), including 188, 123, and 295 subjects, respectively. Each subject has a photo in a frontal pose, normal illumination condition, and neutral expression with a sketch drawn by an artist. Thus these face images naturally come from two views, i.e. the photo and sketch, and we can construct the probe and gallery sets, respectively. We study both the photo vs. sketch and sketch vs. photo settings, while

Table 2: Average rank-one matching accuracy on the CUFS data set (**Best**; Second best).

Matching Type		Photo vs. Sketch			Cropped Photo vs. Original Sketch with $T = 50$			
Training Setting		$T = 25$	$T = 50$	$T = 75$	$m = 2$	$m = 4$	$m = 6$	$m = 8$
CUHK	CCA	38.59 ± 4.89	63.91 ± 4.92	82.65 ± 4.05	71.38 ± 3.42	72.83 ± 2.67	58.26 ± 3.99	33.33 ± 5.04
	2DCCA	74.17 ± 19.26	83.70 ± 12.39	90.88 ± 6.29	61.74 ± 12.21	39.86 ± 14.22	13.91 ± 5.19	8.12 ± 3.07
	MCCA1+2	53.44 ± 8.32	72.10 ± 3.57	82.39 ± 5.13	75.29 ± 2.89	71.38 ± 3.30	57.75 ± 4.81	32.25 ± 4.96
	PCCA	58.65 ± 3.83	80.29 ± 3.58	90.71 ± 3.37	80.65 ± 2.60	78.19 ± 2.76	68.70 ± 2.83	38.55 ± 4.64
	BMTF	96.20 ± 1.15	97.75 ± 0.72	98.41 ± 0.81	69.49 ± 4.31	66.23 ± 7.09	44.35 ± 3.92	25.94 ± 6.31
	BPCCA	**98.71 ± 0.45**	**99.20 ± 0.41**	**99.03 ± 0.65**	**99.20 ± 0.41**	**99.35 ± 0.41**	**99.13 ± 0.57**	**91.52 ± 5.75**
AR	CCA	15.20 ± 4.67	28.77 ± 7.91	48.96 ± 5.31	30.96 ± 4.93	29.45 ± 4.93	24.79 ± 5.97	16.16 ± 3.64
	2DCCA	21.33 ± 5.45	33.15 ± 3.28	45.63 ± 7.25	31.64 ± 5.22	23.97 ± 4.80	16.03 ± 6.20	6.71 ± 3.19
	MCCA1+2	25.00 ± 4.36	39.45 ± 4.02	56.67 ± 5.45	40.68 ± 5.09	38.63 ± 5.24	29.32 ± 8.17	18.22 ± 4.57
	PCCA	23.37 ± 4.45	41.51 ± 6.23	63.13 ± 4.51	41.10 ± 6.36	**40.96 ± 5.97**	32.74 ± 7.49	**22.74 ± 5.09**
	BMTF	28.88 ± 3.54	42.33 ± 7.00	56.88 ± 5.38	22.05 ± 5.34	20.82 ± 4.69	17.12 ± 3.18	9.73 ± 3.79
	BPCCA	**40.71 ± 3.45**	**51.51 ± 7.02**	**70.21 ± 3.26**	**49.04 ± 3.92**	**45.89 ± 5.22**	**40.68 ± 8.47**	**25.48 ± 7.59**



(a) CUHK (b) AR

Figure 1: Cropping examples from CUHK and AR data sets.

the results of the second setting are left in the supplementary material to save space. Since XM2VTS is no longer free for access, it is excluded from our experiments. The CUHK and AR data sets are randomly split so that $T = 25, 50, 75$ faces are selected for training and the rest for test, and the average results over ten such random splits are reported. Each image is resized to 40×32 graylevel pixels.

The left half of Table 2 shows that BPCCA achieves the best performance with statistical significance in all the cases. Other bilinear methods such as 2DCCA and BMTF also obtain good results, especially on the CUHK data set. This indicates that exploiting the matrix structures benefits common feature extraction from heterogeneous images. In addition, bilinear PCCAs outperform their linear counterparts more significantly when T is small, which implies that they are more robust against the small sample size problem.

Robustness on photo-sketch matching: To evaluate the robustness of BPCCA in extracting common features, we use the same data sets and settings for face photo-sketch recognition, while images from *the first view* are cropped by removing $m = 2, 4, 6, 8$ pixels from each of the four sides (see Figure 1), resulting in cropped images of size $(40 - 2m) \times (32 - 2m)$. Images from the second view still have the original size of 40×32. Thus, we are matching images of different sizes. As m increases, photo-sketch pairs gradually loss some important common information such as the face and head outlines, which makes photo-sketch recognition more challenging. Since BMTF can only deal with two-view matrices with the same row dimensions, images of the second view for BMTF are preprocessed by multilinear PCA (Lu, Plataniotis, and Venetsanopoulos 2008) to make the row dimensions of both views equal.

The right half of Table 2 shows the matching performance on the CUFS data set with $T = 50$. For the CUHK student data set, it is clear that BPCCA still performs very well even when m is large, while the performance of other methods is drastically degraded by the photo-sketch mismatch. Specifically, when m varies from 2 to 8, BPCCA degrades only by 1.9% in average. In contrast, BMTF (the 2nd best when $m = 0$) degrades by 46.25% in average. This probably because the model assumptions of BMTF do not hold for images of different sizes, and some critical common information may be lost after dimensionality reduction. For the AR data set, while all the methods obtain relatively poor matching accuracy when $m = 8$ due to severe face-sketch mismatch under large cropping, BPCCA consistently achieves the best performance, and outperforms PCCA (the 2nd best) by 5.89% in average. On the whole, BPCCA achieves good performance, and is more robust against cropping and photo-sketch mismatch.

Convergence study: Finally, we empirically study the convergence property of BPCCA and find that BPCCA is stable and often converges within a relatively small number of iterations. Due to limited space, we put the detailed results and discussions in the supplementary material.

Conclusion

This paper proposed BPCCA, a new bilinear extension of Probabilistic CCA for learning correlations between two-view matrices. Compared with existing bilinear PCCAs, BPCCA is more flexible in capturing two-view correlations and fully enjoys the benefits of the probabilistic framework. By introducing a hybrid joint model with both vector-based and matrix-based concatenations, BPCCA not only preserves the matrix structures with improved model flexibility but also enables parameter estimation with close-form solutions. Experiments on two real-world applications demonstrated the superior performance of BPCCA in matching facial images from different poses and face photos-sketch recognition.

Acknowledgments

This research was supported by Research Grants Council of the Hong Kong SAR (Grant 22200014).

References

Afrabandpey, H.; Safayani, M.; and Mirzaei, A. 2014. Probabilistic two-dimensional canonical correlation analysis for face recognition. In *Proceedings of the 4th International eConference on Computer and Knowledge Engineering*, 1–6.

Bach, F. R., and Jordan, M. I. 2005. A probabilistic interpretation of canonical correlation analysis. Technical Report TR 688, University of California, Berkeley.

Browne, M. W. 1979. The maximum-likelihood solution in inter-battery factor analysis. *British Journal of Mathematical and Statistical Psychology* 32(1):75–86.

Chaudhuri, K.; Kakade, S. M.; Livescu, K.; and Sridharan, K. 2009. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th International Conference on Machine Learning*, 129–136. ACM.

Damianou, A.; Lawrence, N. D.; and Ek, C. H. 2016. Multi-view learning as a nonparametric nonlinear inter-battery factor analysis. *arXiv preprint arXiv:1604.04939*.

Gang, L.; Yong, Z.; Yan-Lei, L.; and Jing, D. 2011. Three dimensional canonical correlation analysis and its application to facial expression recognition. In *Intelligent Computing and Information Science*. Springer. 56–61.

Gupta, A. K., and Nagar, D. K. 1999. *Matrix Variate Distributions*, volume 104. CRC Press.

Hardoon, D. R.; Szedmak, S.; and Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation* 16(12):2639–2664.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28(3-4):321–377.

Khan, S. A., and Kaski, S. 2014. Bayesian multi-view tensor factorization. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 656–671.

Klami, A.; Virtanen, S.; and Kaski, S. 2013. Bayesian canonical correlation analysis. *Journal of Machine Learning Research* 14(Apr):965–1003.

Lee, S. H., and Choi, S. 2007. Two-dimensional canonical correlation analysis. *IEEE Signal Processing Letters* 14(10):735.

Leurgans, S. E.; Moyeed, R. A.; and Silverman, B. W. 1993. Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society. Series B (Methodological)* 55(3):725–740.

Lu, H.; Plataniotis, K. N.; and Venetsanopoulos, A. N. 2008. MPCA: Multilinear principal component analysis of tensor objects. *IEEE Transaction on Neural Networks* 19(1):18–39.

Lu, H.; Plataniotis, K. N.; and Venetsanopoulos, A. N. 2013. *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. CRC Press.

Lu, H. 2013. Learning canonical correlations of paired tensor sets via tensor-to-vector projection. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 1516–1522.

Martinez, A. M. 1998. The AR face database. Technical Report CVC 24, Computer Vision Center.

Meng, X., and Rubin, D. B. 1993. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* 80(2):267–278.

Messer, K.; Matas, J.; Kittler, J.; Luettin, J.; and Maitre, G. 1999. XM2VTSDB: The extended M2VTS database. In *Proceedings of the 2nd International Conference on Audio and Video-Based Biometric Person Authentication*, 72–77.

Sim, T.; Baker, S.; and Bsat, M. 2003. The CMU pose, illumination, and expression database. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 25(12):1615–1618.

Tang, X., and Wang, X. 2003. Face sketch synthesis and recognition. In *Proceedings of 9th IEEE International Conference on Computer Vision*, 687–694. IEEE.

Vinod, H. D. 1976. Canonical ridge and econometrics of joint production. *Journal of Econometrics* 4(2):147–166.

Weenink, D. 2003. Canonical correlation analysis. In *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, volume 25, 81–99. Citeseer.

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

Yan, J.; Zheng, W.; Zhou, X.; and Zhao, Z. 2012. Sparse 2-d canonical correlation analysis via low rank matrix approximation for feature extraction. *IEEE Signal Processing Letters* 19(1):51–54.

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

Zhang, Y., and Schneider, J. G. 2011. Multi-label output codes using canonical correlation analysis. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 873–882.

Zhao, J.; Yu, P. L. H.; and Kwok, J. T. 2012. Bilinear probabilistic principal component analysis. *IEEE Transaction on Neural Networks and Learning Systems* 23(3):492–503.