Probabilistic Rank-One Discriminant Analysis via Collective and Individual Variation Modeling

Yang Zhou and Yiu-ming Cheung, *Fellow, IEEE*

***Abstract*—Linear discriminant analysis (LDA) is a classical supervised subspace learning technique that has wide applications. However, it is designed for vector only, which cannot exploit the tensor structures and may lead to suboptimal results when dealing with tensorial data. To address this problem, several multilinear LDA (MLDA) methods have been proposed to learn the subspaces from tensors. By exploiting the tensor structures, they achieve compact subspace representations, reduced parameter sizes, and improved robustness against the small sample size problem. However, existing MLDA methods do not take data uncertainty into account, fail to converge properly, or have to introduce additional tuning parameters for good convergence properties. In this paper, we therefore solve these limitations by proposing a probabilistic MLDA method for matrix inputs. Specifically, we propose a new generative model to incorporate structural information into the probabilistic framework, where each observed matrix is represented as a linear combination of collective and individual rank-one matrices. This provides our method with both the expressiveness of capturing discriminative features and nondiscriminative noise, and the capability of exploiting the 2-D tensor structures. To overcome the convergence problem of existing MLDAs, we develop an EM-type algorithm for parameter estimation, which has closed-form solutions with convergence guarantees. Experimental results on real-world datasets show the superiority of the proposed method to other probabilistic and MLDA variants.**

***Index Terms*—Discriminant analysis, multilinear subspace learning, probabilistic models.**

I. Introduction

DUE TO the difficulties of processing high-dimensional data, the demand for dimensionality reduction is pervasive in pattern recognition, data mining, computer vision, and so on [1]–[5]. Subspace learning techniques aim at representing high-dimensional data in a low-dimensional subspace while preserving their intrinsic characteristics. By handling data in the learned subspace, subsequent tasks such as clustering, classification, visualization, and interpretation can be greatly facilitated. Principal component analysis (PCA) [6] and linear discriminant analysis (LDA) [7] are probably the most popular subspace learning techniques. PCA is an unsupervised method, which learns a subspace that preserves maximum data variance. LDA is a supervised method. It learns a subspace by maximizing between-class scatter while minimizing within-class scatter, so that data of the same class are grouped together while those of different classes are well separated. Because of its supervised nature, LDA is generally more suitable than PCA for dealing with labeled data.

In real-life, many data are naturally organized as *tensors* [8]–[11]. For example, gray-level images are 2-D tensors (matrices), gray-scale video sequences are 3-D tensors, and fMRI images are 4-D tensors. Although, the effectiveness of LDA has been demonstrated in many applications, it is designed for *vectors* only, and has to first reshape tensorial inputs into vectors when it comes to dealing with tensors. Consequently, LDA fails to exploit the structural information from tensors due to the reshaping. Moreover, it is often the case that many real-world tensors, such as whole-brain MRI or fMRI scans are very high-dimensional, where the number of samples is usually much smaller than that of features. In this scenario, LDA may suffer from the small sample size (SSS) problem and obtain degraded performance [12] because the estimation of between- and within-class scatter tends to be inaccurate with limited sample sizes.

To address these problems, several multilinear LDA (MLDA) methods have been proposed, which are motivated by the observations that tensor structures are very informative and helpful for subspace learning, and can be used to alleviate the SSS problem [13]–[17]. By exploiting the tensor structures, MLDAs learn multilinear projections to reduce the dimensionality of tensors from each direction (i.e., mode), e.g., the column and row directions for 2-D tensors (matrices). This provides MLDAs with compact subspace representations, reduced parameter sizes, and improved robustness in estimating the scatter matrices [18].

2-D LDA (2DLDA) [19] takes matrices as inputs, and alternately learns the column and row subspaces by maximizing the ratio of between-class to within-class scatter. Discriminant analysis with tensor representation (DATER) [20] generalizes 2DLDA to higher-order cases for dealing with general tensors. Based on the same scatter-ratio-based discriminant criterion, uncorrelated MLDA (UMLDA) [21] further imposes

Manuscript received May 31, 2018; revised August 7, 2018; accepted September 11, 2018. Date of publication October 4, 2018; date of current version December 16, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61672444 and Grant 61272366, in part by the Faculty Research Grant of Hong Kong Baptist University under Project FRG2/17-18/082 and Project FRG2/16-17/051, in part by KTO Grant of HKBU under Project MPCF-004-2017/18, and in part by SZSTI under Grant JCYJ20160531194006833. This paper was recommended by Associate Editor D. Tao. *(Corresponding author: Yiu-ming Cheung.)*

Y. Zhou is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: yangzhou@comp.hkbu.edu.hk).

Y.-M. Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, and also with the Institute of Research and Continuing Education, Hong Kong Baptist University, Hong Kong (e-mail: ymc@comp.hkbu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2870440

uncorrelated constraints on subspace bases for less feature redundancy, where each basis is solved in a greedy and successive way.

One main limitation of the above ratio-based MLDAs is that they fail to monotonically increase their objective function, i.e., the scatter ratio, over iterations, and thus have *no convergence guarantee*. To avoid this problem, general tensor discriminant analysis (GTDA) [22] learns subspaces by maximizing the scatter *difference* rather than the *ratio*. Similarly, tensor rank-one discriminant analysis (TR1DA) [23] successively finds each subspace basis based on the scatter difference criterion. Unlike their ratio-based counterparts, difference-based MLDAs can monotonically increase the scatter difference over iterations, and thus achieve good convergence properties. However, they have to introduce *additional tuning parameters* to control the weight between the between- and within-class scatter. From the practical perspective, these parameters are often sensitive and difficult to be well-determined.

Although, both the scatter ratio and difference are valid discriminant criteria, they suffer from their own limitations for *multilinear* subspace learning, leading to either convergence problems or additional tuning parameters. To address these limitations, this paper, therefore, considers multilinear subspace learning from a probabilistic perspective, and proposes a probabilistic MLDA for matrix inputs, named as probabilistic rank-one discriminant analysis (PRODA).

Instead of taking the scatter ratio or difference as the discriminant criterion, PRODA aims at maximizing the *log-likelihood* of a generative model that characterizes between- and within-class information by the variation of collective and individual latent features, respectively. In this way, PRODA inherits the capability of MLDAs in exploiting the 2-D tensor structures while achieving guaranteed convergence without introducing additional tuning parameters. Moreover, it is well-known that the probabilistic framework offers unique advantages in capturing data uncertainty, handling missing data, and Bayesian model selection. Theses benefits also motivate us to develop a probabilistic approach to MLDAs accordingly. Our contribution is summarized below.

1) We propose a new generative model for learning discriminative subspaces from matrices, where each observed matrix is represented as a linear combination of collective and individual rank-one matrices. In this way, the proposed model achieves both the expressiveness of capturing discriminative features and nondiscriminative noise, and the capability of exploiting the 2-D tensor structures.
2) To overcome the convergence problem of existing MLDAs, we develop an EM-type algorithm for parameter estimation. By maximizing the log-likelihood rather than the scatter ratio or difference, the proposed method is guaranteed to converge to local optima without introducing additional tuning parameters.

It is worth noting that PRODA is designed for 2-D tensors (matrices), but the proposed idea is general and can be extended to higher-order cases.

The rest of this paper is organized as follows. Section II introduces the notations used in this paper and briefly reviews some related work. The proposed PRODA method is then presented in Section III. Section IV evaluates PRODA on real-world datasets against the competing LDA variants, and finally Section V gives the concluding remarks.

II. Preliminaries

A. Notations

Vectors are denoted by boldface lowercase letters, e.g., $\mathbf{x}$. Matrices are denoted by boldface capital letters, e.g., $\mathbf{X}$. Symbols $\otimes$, $\odot$, and $\circledast$ denote the Kronecker, column-wise Kronecker, and entry-wise products, respectively. $\langle\cdot\rangle$ denotes the expectation with respect to a certain distribution. $\text{vec}(\mathbf{X})$ is the vector stacked by the columns of $\mathbf{X}$. $\text{diag}(\mathbf{x})$ is the diagonal matrix created from $\mathbf{x}$. The random matrix $\mathbf{X} \in \mathbb{R}^{d_c \times d_r} \sim \mathcal{N}_{d_c,d_r}(\Xi, \Sigma_c, \Sigma_r)$ means that $\mathbf{X}$ follows the *matrix-variate normal distribution* with the mean matrix $\Xi \in \mathbb{R}^{d_c \times d_r}$, the column covariance matrix $\Sigma_c \in \mathbb{R}^{d_c \times d_c}$, and the row covariance matrix $\Sigma_r \in \mathbb{R}^{d_r \times d_r}$.

B. Linear Discriminant Analysis

Let $\{\{\mathbf{x}_{jk} \in \mathbb{R}^d\}_{j=1}^{N_k}\}_{k=1}^{K}$ be the training set that consists of $N = \sum_{k=1}^{K} N_k$ samples from K classes, where $\mathbf{x}_{jk}$ is the jth sample of the kth class, and N_k is the number of samples in the kth class. LDA seeks a linear projection $\mathbf{U}^{(\text{LDA})} \in \mathbb{R}^{d \times q}$ that maximizes the Fisher's discriminant ratio as follows [7]:

$$\mathbf{U}^{(\text{LDA})} = \arg\max_{\mathbf{U}} \frac{\left|\mathbf{U}^\top \mathbf{S}_B^{(\text{LDA})} \mathbf{U}\right|}{\left|\mathbf{U}^\top \mathbf{S}_W^{(\text{LDA})} \mathbf{U}\right|} \tag{1}$$

where $\mathbf{S}_B^{(\text{LDA})} \in \mathbb{R}^{d \times d}$ and $\mathbf{S}_W^{(\text{LDA})} \in \mathbb{R}^{d \times d}$ are the between- and within-class scatter matrices, respectively.

The definitions of $\mathbf{S}_B^{(\text{LDA})}$ and $\mathbf{S}_W^{(\text{LDA})}$ are given by

$$\mathbf{S}_B^{(\text{LDA})} = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\top \tag{2}$$

$$\mathbf{S}_W^{(\text{LDA})} = \sum_{jk} (\mathbf{x}_{jk} - \boldsymbol{\mu}_k)(\boldsymbol{x}_{jk} - \boldsymbol{\mu}_k)^\top \tag{3}$$

where $\sum_{jk}$ is the abbreviation of $\sum_{k=1}^{K}\sum_{j=1}^{N_k}$, $\boldsymbol{\mu} = (1/N)\sum_{jk}\boldsymbol{x}_{jk}$ is the mean of the whole training set, and $\boldsymbol{\mu}_k = (1/N_k)\sum_{j=1}^{N_k}\boldsymbol{x}_{jk}$ is the mean of the kth class. The LDA solution $\mathbf{U}^{(\text{LDA})}$ is given by the eigenvectors of $(\mathbf{S}_W^{(\text{LDA})})^{-1}\mathbf{S}_B^{(\text{LDA})}$ corresponding to the q largest eigenvalues.

C. Probabilistic LDA

LDA establishes a simple and effective way of supervised subspace learning, and has been extended for different applications. Among various LDA extensions, Probabilistic LDA (PLDA) [24]–[26] is one of the most popular representatives, which is closely related to our PRODA method. Unlike LDA, PLDA learns discriminative subspaces by estimating a generative model that characterizes between-class and within-individual variation. Specifically, PLDA models the jth observed vector of the kth class $\mathbf{x}_{jk}$ as follows:

$$\mathbf{x}_{jk} = \mathbf{U}_y \mathbf{y}_k + \mathbf{U}_z \mathbf{z}_{jk} + \boldsymbol{\mu} + \boldsymbol{\epsilon}_{jk} \tag{4}$$

where $\mathbf{y}_k$ is the P_y-dimensional latent class variable, $\mathbf{z}_{jk}$ is the P_z-dimensional latent individual variable, $\mathbf{U}_y \in \mathbb{R}^{d\times P_y}$ is the class factor matrix, $\mathbf{U}_z \in \mathbb{R}^{d\times P_z}$ is the individual factor matrix, $\boldsymbol{\epsilon}_{jk} \in \mathbb{R}^d \sim \mathcal{N}(\mathbf{0}, \Lambda)$ is the random noise with the diagonal covariance matrix Λ, and $\boldsymbol{\mu}$ is the mean vector.

The PLDA model (4) can be divided into two parts: 1) the discriminative part $\mathbf{U}_y\mathbf{y}_k + \boldsymbol{\mu}$ that is shared by all the observations of the kth class and describes between-class variation and 2) the noise part $\mathbf{U}_z\mathbf{z}_{jk} + \boldsymbol{\epsilon}_{jk}$ that is different for individual observations and represents within-individual variation. Since PLDA explicitly characterizes both the class and noise components, it takes data uncertainty and individual-specific variation into account, and thus can extract discriminative features that may be discarded or considered as less important by LDA [24].

1) More Probabilistic LDA Variants: Besides PLDA, there are also some other probabilistic LDA variants. Ioffe [27] possibly proposed the first probabilistic extension of LDA. This approach models between- and within-class scatter in Gaussian distributions, which eventually results in a *weighted form* of the classical LDA solution. Ioffe's probabilistic LDA assumes that each class can only consist of the *same* number of training samples. Such assumption is usually impractical, and greatly limits the effectiveness and applicability of Ioffe's probabilistic LDA.

Yu *et al.* [28] proposed a supervised PCA method, which performs discriminant analysis by maximizing the correlation between each observation and its class indicator vector. Although, it is derived from the probabilistic perspective, the maximum likelihood solution of this supervised PCA is just *identical* to that of the classical LDA. Apart from the above probabilistic LDA variants, some attempts have also been made to utilize the probabilistic framework for heterogeneous face recognition [29] and data restoration [30].

2) Neglected Tensor Structures: All the above-mentioned LDA variants are designed for vectors only, while many real-world data such as images and videos are naturally in the form of tensors rather than vectors. Under the circumstances, when it comes to dealing with tensorial data, these vector-based LDAs have to first reshape tensors into vectors, which breaks the tensor structures. Consequently, they fail to discover the structural information from tensors, and may lead to suboptimal results for certain applications [31], [32].

D. Multilinear LDA

To exploit the tensor structures for discriminant analysis, several MLDA methods have been proposed. According to different discriminant criteria used in the subspace learning, they can be grouped into two categories: 1) ratio-based and 2) difference-based MLDAs. For clarity, we introduce them in 2-D cases, where the inputs are *matrices*.

1) Ratio-Based MLDAs: Let $\{\{\mathbf{X}_{jk} \in \mathbb{R}^{d_c\times d_r}\}_{j=1}^{N_k}\}_{k=1}^{K}$ be the training set, where $\mathbf{X}_{jk}$ is the jth matrix input of the kth class. Ratio-based MLDAs aim at finding multilinear projections that maximize the *ratio* of between-class to within-class scatter. For instance, 2DLDA [19] learns two projection matrices $\mathbf{U}_c \in \mathbb{R}^{d_c\times q_c}$ and $\mathbf{U}_r \in \mathbb{R}^{d_r\times q_r}$, which characterize the column and row subspaces, respectively. Those projection matrices are solved iteratively and alternately based on the following scatter ratio criterion. By fixing $\mathbf{U}_r$, $\mathbf{U}_c$ is solved by:

$$\mathbf{U}_c^{(\text{2DLDA})} = \arg\max_{\mathbf{U}} \frac{\operatorname{tr}\left(\mathbf{U}^\top\mathbf{S}_{B_c}^{(\text{MLDA})}\mathbf{U}\right)}{\operatorname{tr}\left(\mathbf{U}^\top\mathbf{S}_{W_c}^{(\text{MLDA})}\mathbf{U}\right)} \tag{5}$$

where $\mathbf{S}_{B_c}^{(\text{MLDA})} \in \mathbb{R}^{d_c\times d_c}$ and $\mathbf{S}_{W_c}^{(\text{MLDA})} \in \mathbb{R}^{d_c\times d_c}$ are the *column-wise* between- and within-class scatter matrices, respectively. The definitions of $\mathbf{S}_{W_c}^{(\text{MLDA})}$ and $\mathbf{S}_{B_c}^{(\text{MLDA})}$ are as follows:

$$\mathbf{S}_{B_c}^{(\text{MLDA})} = \sum_{k=1}^{K} N_k(\mathbf{M}_k - \mathbf{M})\mathbf{U}_r\mathbf{U}_r^\top(\mathbf{M}_k - \mathbf{M})^\top \tag{6}$$

$$\mathbf{S}_{W_c}^{(\text{MLDA})} = \sum_{jk}\left(\mathbf{X}_{jk} - \mathbf{M}_k\right)\mathbf{U}_r\mathbf{U}_r^\top\left(\mathbf{X}_{jk} - \mathbf{M}_k\right)^\top \tag{7}$$

where $\mathbf{M} = (1/N)\sum_{jk}\mathbf{X}_{jk}$ is the overall mean matrix and $\mathbf{M}_k = (1/N_k)\sum_{j=1}^{N_k}\mathbf{X}_{jk}$ is the class mean matrix. Analogous to LDA, the solution $\mathbf{U}_c^{(\text{2DLDA})}$ is given by the eigenvectors of $(\mathbf{S}_{W_c}^{(\text{MLDA})})^{-1}\mathbf{S}_{B_c}^{(\text{MLDA})}$ associated with the q_c largest eigenvalues.

By fixing $\mathbf{U}_c$, the row projection $\mathbf{U}_r^{(\text{2DLDA})}$ is solved by

$$\mathbf{U}_r^{(\text{2DLDA})} = \arg\max_{\mathbf{U}} \frac{\operatorname{tr}\left(\mathbf{U}^\top\mathbf{S}_{B_r}^{(\text{MLDA})}\mathbf{U}\right)}{\operatorname{tr}\left(\mathbf{U}^\top\mathbf{S}_{W_r}^{(\text{MLDA})}\mathbf{U}\right)}. \tag{8}$$

$\mathbf{S}_{B_r}^{(\text{MLDA})} \in \mathbb{R}^{d_r\times d_r}$ and $\mathbf{S}_{W_r}^{(\text{MLDA})} \in \mathbb{R}^{d_r\times d_r}$ are the *row-wise* between- and within-class scatter matrices, respectively, whose definitions are given by

$$\mathbf{S}_{B_r}^{(\text{MLDA})} = \sum_{k=1}^{K} N_k(\mathbf{M}_k - \mathbf{M})^\top\mathbf{U}_c\mathbf{U}_c^\top(\mathbf{M}_k - \mathbf{M}) \tag{9}$$

$$\mathbf{S}_{W_r}^{(\text{MLDA})} = \sum_{jk}\left(\mathbf{X}_{jk} - \mathbf{M}_k\right)^\top\mathbf{U}_c\mathbf{U}_c^\top\left(\mathbf{X}_{jk} - \mathbf{M}_k\right). \tag{10}$$

Then $\mathbf{U}_r^{(\text{2DLDA})}$ is given by the eigenvectors of $(\mathbf{S}_{W_r}^{(\text{MLDA})})^{-1}\mathbf{S}_{B_r}^{(\text{MLDA})}$ associated with the q_r largest eigenvalues.

With the same objective function, DATER [20] generalizes 2DLDA to higher-order cases, which can deal with general tensors. UMLDA [21] is also designed for general tensors and employs the ratio-based discriminant criterion, while it finds each subspace basis (column of $\mathbf{U}_c$ and $\mathbf{U}_r$) in a greedy and successive way. Moreover, UMLDA imposes uncorrelated constraints on the projection matrices to extract independent features and reduce the subspace redundancy.

2) Difference-Based MLDAs: Difference-based MLDAs learn discriminative subspaces by maximizing the *difference* between between- and within-class scatter. For instance, GTDA [22] learns the column and row projections by alternately maximizing the following two objective functions:

$$\mathbf{U}_c^{(\text{GTDA})} = \arg\max_{\mathbf{U}} \operatorname{tr}\left(\mathbf{U}^\top\left(\mathbf{S}_{B_c}^{(\text{MLDA})} - \xi_c\mathbf{S}_{W_c}^{(\text{MLDA})}\right)\mathbf{U}\right) \tag{11}$$

$$\mathbf{U}_r^{(\text{GTDA})} = \arg\max_{\mathbf{U}} \operatorname{tr}\left(\mathbf{U}^\top\left(\mathbf{S}_{B_r}^{(\text{MLDA})} - \xi_r\mathbf{S}_{W_r}^{(\text{MLDA})}\right)\mathbf{U}\right) \tag{12}$$

where ξ_c and ξ_r are the tuning parameters whose values are heuristically set to the largest eigenvalue of $(\mathbf{S}_{W_c}^{(\mathrm{MLDA})})^{-1}\mathbf{S}_{B_c}^{(\mathrm{MLDA})}$ and $(\mathbf{S}_{W_r}^{(\mathrm{MLDA})})^{-1}\mathbf{S}_{B_r}^{(\mathrm{MLDA})}$, respectively. With fixed $\mathbf{U}_r$, the solution $\mathbf{U}_c^{(\mathrm{GTDA})}$ is given by the eigenvectors of $\mathbf{S}_{B_c}^{(\mathrm{MLDA})} - \xi_c\mathbf{S}_{W_c}^{(\mathrm{MLDA})}$ associated with the q_c largest eigenvalues. The row projection $\mathbf{U}_r^{(\mathrm{GTDA})}$ can be solved in a similar way. Along this line, TR1DA [23] successively finds each column of $\mathbf{U}_c$ and $\mathbf{U}_r$ based on the scatter difference criterion, which can be viewed as a difference-based version of UMLDA without the uncorrelated constraints.

3) Exploited Structural Information: In general, the performance of LDA is highly dependent on the quality of the scatter matrices $\mathbf{S}_B^{(\mathrm{LDA})}$ and $\mathbf{S}_W^{(\mathrm{LDA})}$. When dealing with real-world tensors such as whole-brain MRI or fMRI scans, it is often the case that the number of training samples is much smaller than that of input features. In this scenario, the estimation of between- and within-class scatter tends to be inaccurate due to the limited sample size. As a result, the performance of LDA could be seriously degraded, which is known as the SSS problem [12], [33]–[35].

By exploiting the tensor structures, MLDAs can extract discriminative information from the *multilinear* scatter matrices such as $\mathbf{S}_{B_c}^{(\mathrm{MLDA})}$ (6) and $\mathbf{S}_{W_c}^{(\mathrm{MLDA})}$ (7), which have much smaller sizes and better conditioning than the original ones (2) and (3). Because of this, MLDAs save much memory cost for storing the scatter matrices, reduce the parameter size of LDA from dq for $\mathbf{W}$ to $d_cq_c + d_rq_r$ for $\mathbf{U}_c$ and $\mathbf{U}_r$, and most importantly gain the robustness in estimating between- and within-class scatter from SSSs.

4) Convergence and Tuning Parameter Issues: One great limitation of ratio-based MLDAs is that they fail to monotonically increase their objective functions, i.e., the column- and row-wise scatter ratios (5) and (8), over iterations, and may not converge properly [14], [36]–[38]. This is because the column-wise solution for (5) does *not* necessarily increase the row-wise scatter ratio (8) and vice versa. Therefore, the results of ratio-based MLDAs would be unstable during iterations, and have no convergence guarantee. In contrast, difference-based MLDAs avoid the convergence problem, since the scatter differences (11) and (12) can be monotonically increased in the alternate optimization procedure. However, they have to introduce the tuning parameters ξ_c and ξ_r for controlling the weight between the between- and within-class scatter, which are often sensitive and difficult to be well-determined in practice.

III. Proposed Method

Although, PLDA and MLDA enjoy some benefits over LDA, they still have their own limitations. This section presents a probabilistic MLDA for matrix inputs, named as PRODA. Different from existing MLDAs that employ the scatter ratio or difference, PRODA takes the log-likelihood of a generative model as the discriminant criterion, where between- and within-class information is characterized by the variation of collective and individual latent features, respectively. In this way, it inherits the benefits of both PLDA and MLDA while avoiding their limitations.

In the following, PRODA is presented in three stages.

1) We first propose the PRODA model to incorporate the matrix structures into the probabilistic framework.
2) An EM-type algorithm is then developed for parameter estimation to overcome the convergence problem of MLDAs, where the *M*-step is further regularized for better generalization.
3) Finally, we discuss the initialization strategies and analyze the time complexity of PRODA.

A. PRODA Model

Analogous to the singular value decomposition, we decompose the jth observed matrix of the kth class $\mathbf{X}_{jk}$ into a number of *rank-one matrices* as follows:

$$\begin{aligned}\mathbf{X}_{jk} &= \sum_{p=1}^{P_y} y_p^k \mathbf{c}_p^y \mathbf{r}_p^{y\top} + \sum_{q=1}^{P_z} z_q^{jk} \mathbf{c}_q^z \mathbf{r}_q^{z\top} + \mathbf{E}_{jk} \\ &= \mathbf{C}_y \ \mathrm{diag}(\mathbf{y}_k)\mathbf{R}_y^\top + \mathbf{C}_z \ \mathrm{diag}\big(\mathbf{z}_{jk}\big)\mathbf{R}_z^\top + \mathbf{E}_{jk} \qquad (13)\end{aligned}$$

where $\mathbf{y}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{P_y})$ is the P_y-dimensional collective latent variable, $\mathbf{z}_{jk} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{P_z})$ is the P_z-dimensional individual latent variable, and $\mathbf{E}_{jk} \sim \mathcal{N}_{d_c,d_r}(\mathbf{0}, \sigma\mathbf{I}, \sigma\mathbf{I})$ is the random noise matrix with $\sigma > 0$. $\mathbf{C}_{y/z} \in \mathbb{R}^{d_c \times P_{y/z}}$ and $\mathbf{R}_{y/z} \in \mathbb{R}^{d_r \times P_{y/z}}$ are the collective/individual column and row factor matrices, respectively.

The PRODA model represents each observation as a linear combination of $P_y + P_z$ rank-one matrices. Among them, P_y of them are discriminative factors that characterize between-class variation, while the others are nondiscriminative ones that describe within-individual variation. The coefficients $\mathbf{y}_k$ and $\mathbf{z}_{jk}$ serve as the low-dimensional latent representations in the discriminative and individual subspaces, respectively. $\mathbf{y}_k$ can be viewed as the class identity, and is shared by all the observations of the kth class. On the other hand, different $\mathbf{z}_{jk}$s are independent of each other, and can be viewed as structured noise.

Besides the capability of capturing discriminative features and nondiscriminative noise, there is another key benefit can be obtained by representing each observation as a number of rank-one matrices. Specifically, the PRODA model (13) can naturally group the *collective* and *individual* rank-one matrices together in a *joint* form as follows:

$$\begin{aligned}\mathbf{X}_{jk} &= \big[\mathbf{C}_y, \mathbf{C}_z\big]\begin{bmatrix}\mathrm{diag}(\mathbf{y}_k) & \mathbf{0} \\ \mathbf{0} & \mathrm{diag}(\mathbf{z}_{jk})\end{bmatrix}\big[\mathbf{R}_y, \mathbf{R}_z\big]^\top + \mathbf{E}_{jk} \\ &= \mathbf{C} \ \mathrm{diag}\big(\mathbf{f}_{jk}\big)\mathbf{R}^\top + \mathbf{E}_{jk} \qquad (14)\end{aligned}$$

where $\mathbf{f}_{jk} = [\mathbf{y}_k^\top, \mathbf{z}_{jk}^\top]^\top$ is the joint latent variable, $\mathbf{C} = [\mathbf{C}_y, \mathbf{C}_z]$ and $\mathbf{R} = [\mathbf{R}_y, \mathbf{R}_z]$ are the column and row factor matrices, respectively. By combining the collective and individual factors in $\mathbf{C}$ and $\mathbf{R}$, such joint form not only preserves the spatial structures of $\mathbf{X}_{jk}$ but also greatly facilitates the subsequent parameter estimation, leading to closed-form solutions with guaranteed convergence. Fig. 1 gives the graphical model for PRODA.

Armed with the PRODA model (14), we can obtain the conditional distribution $p(\mathbf{X}_{jk}|\mathbf{f}_{jk})$ as follows:

$$\mathbf{X}_{jk}\Big|\mathbf{f}_{jk} \sim \mathcal{N}_{d_c,d_r}\Big(\mathbf{C} \ \mathrm{diag}\big(\mathbf{f}_{jk}\big)\mathbf{R}^\top, \sigma\mathbf{I}, \sigma\mathbf{I}\Big). \qquad (15)$$

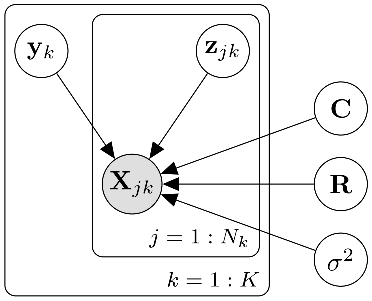


Fig. 1. Graphical model for PRODA.

Using the properties of the column-wise Kronecker product, (15) can be rewritten in the following vector form:

$$\mathbf{x}_{jk}|\mathbf{f}_{jk} \sim \mathcal{N}\Big(\mathbf{W}\mathbf{f}_{jk}, \sigma^2\mathbf{I}\Big) \tag{16}$$

where $\mathbf{x}_{jk} = \text{vec}(\mathbf{X}_{jk})$, $\mathbf{W}_y = \mathbf{R}_y \odot \mathbf{C}_y$, $\mathbf{W}_z = \mathbf{R}_z \odot \mathbf{C}_z$, $\mathbf{W} = \mathbf{R} \odot \mathbf{C} = [\mathbf{W}_y, \mathbf{W}_z]$. In the following derivation, we need both the vector and matrix forms of the conditional distribution to obtain tractable formulations for the log-likelihood and posterior expectations.

Given the training set $\{\{\mathbf{X}_{jk}\}_{j=1}^{N_k}\}_{k=1}^{K}$, our aim is to find the model parameter set $\boldsymbol{\theta} = \{\mathbf{C}, \mathbf{R}, \sigma^2\}$ that maximizes the following log-likelihood function:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_{jk} \ln p\big(\mathbf{X}_{jk}, \mathbf{f}_{jk}\big) \\ &= -\sum_{jk}\bigg[\frac{d_c d_r}{2}\ln\sigma^2 + \frac{1}{2}\mathbf{f}_{jk}^{\top}\mathbf{f}_{jk} + \frac{1}{2\sigma^2}\Big\|\mathbf{X}_{jk} \\ &\qquad - \mathbf{C}\ \text{diag}(\mathbf{f}_{jk})\mathbf{R}^{\top}\Big\|_F^2\bigg] + \text{const.}\end{aligned} \tag{17}$$

This can be achieved by learning the joint latent variable $\mathbf{f}_{jk}$ and the model parameters $\boldsymbol{\theta}$ sequentially under the EM framework.

1) Comparison With PLDA: Both PLDA and PRODA construct a generative model to characterize collective and individual variation. However, PRODA achieves this by representing each observation as a linear combination of *rank-one matrices* rather than *vectors*. This leads to the following advantages over PLDA.

1) PRODA can preserve the spatial structures of the observed matrices, which could be utilized to improve the performance of discriminant analysis.
2) PRODA has compact subspace representations with fewer model parameters to be estimated. Specifically, PLDA has $d_c d_r(P_y + P_z)$ parameters for $\mathbf{W}_y$ and $\mathbf{W}_z$, while PRODA has $(d_c + d_r)(P_y + P_z)$ ones for $\mathbf{C}$ and $\mathbf{R}$.
3) The reduced parameter size (model complexity) in turn could improve the robustness of PRODA in the parameter estimation with SSSs.

2) Comparison With MLDAs: To the best of our knowledge, PRODA is the first bilinear probabilistic LDA, which takes advantages of not only the *matrix structures* for robustness against the SSS problem but also the *probabilistic framework* for compact and flexible subspace representations. In addition, the objective function of PRODA is the *log-likelihood* (17) rather than the *scatter ratio* or *difference*. As will be seen in the next section, it can be monotonically increased under the EM framework, providing PRODA with guaranteed convergence. In contrast, ratio-based MLDAs such as DATER and UMLDA fail to monotonically increase the scatter ratio over iterations, and thus have no convergence guarantee. Although difference-based MLDAs such as GTDA and TR1DA avoid the convergence problem, they involve additional tuning parameters, which are often sensitive and difficult to be well-determined in practice.

B. Parameter Estimation for PRODA

This section develops an EM-type algorithm for estimating $\boldsymbol{\theta}$, which has guaranteed convergence without introducing additional tuning parameters. Since $\mathbf{C}$ and $\mathbf{R}$ are coupled together, it is difficult to optimize them simultaneously. We solve this problem by using the expectation-conditional maximization (ECM) approach [39]. Our ECM algorithm optimizes $\mathbf{C}$ and $\mathbf{R}$ conditionally, and consists of two stages: the *E*-step and CM-step.

1) E-Step: In the *E*-step, the joint latent variable $\mathbf{f}_{jk}$ is optimized with respect to $\mathcal{L}(\boldsymbol{\theta})$ given $\boldsymbol{\theta}$. The solution turns out to be the expectations of $\mathbf{f}_{jk}$ in terms of the posterior $p(\mathbf{f}_{jk}|\mathbf{X}_k)$, where $\mathbf{X}_k = [\mathbf{x}_{1k}, \ldots, \mathbf{x}_{N_k k}]$. Since the collective latent variable $\mathbf{y}_k$ is determined by all the observations of the kth class while the individual latent variable $\mathbf{z}_{jk}$ is only related to the corresponding observation $\mathbf{x}_{jk}$, we decompose the posterior $p(\mathbf{f}_{jk}|\mathbf{X}_k)$ into *two factors*: $p(\mathbf{f}_{jk}|\mathbf{X}_k) = \prod_{j=1}^{N_k} p(\mathbf{z}_{jk}|\mathbf{y}_k, \mathbf{x}_{jk})p(\mathbf{y}_k|\mathbf{X}_k)$ and derive them separately.

Outer Posterior: Given $\mathbf{y}_k$, the individual latent variables $\mathbf{z}_{jk}$ ($j = 1, \ldots, N_k$) of the kth class are conditionally independent of each other. Applying the Gaussian properties, we can obtain the outer posterior $p(\mathbf{z}_{jk}|\mathbf{x}_{jk}, \mathbf{y}_k)$ as follows:

$$\mathbf{z}_{jk}|\mathbf{x}_{jk}, \mathbf{y}_k \sim \mathcal{N}\Big(\mathbf{M}_z\mathbf{W}_z^{\top}\big(\mathbf{x}_{jk} - \mathbf{W}_y\mathbf{y}_k\big), \sigma^2\mathbf{M}_z\Big) \tag{18}$$

where $\mathbf{M}_z = (\mathbf{W}_z^{\top}\mathbf{W}_z + \sigma^2\mathbf{I})^{-1}$.

Inner Posterior: According to Bayes' rule, we have

$$\begin{aligned}p(\mathbf{y}_k|\mathbf{X}_k) &= \frac{p(\mathbf{X}_k, \mathbf{y}_k)}{p(\mathbf{X}_k)} = \frac{p(\mathbf{X}_k, \mathbf{y}_k, \mathbf{Z}_k)}{p(\mathbf{Z}_k|\mathbf{X}_k, \mathbf{y}_k)p(\mathbf{X}_k)} \\ &= \prod_{j=1}^{N_k}\frac{p\big(\mathbf{x}_{jk}|\mathbf{f}_{jk}\big)p(\mathbf{y}_k)p\big(\mathbf{z}_{jk}\big)}{p\big(\mathbf{z}_{jk}|\mathbf{x}_{jk}, \mathbf{y}_k\big)p(\mathbf{x}_{jk})}\end{aligned} \tag{19}$$

where $\mathbf{Z}_k = [\mathbf{z}_{1k}, \ldots, \mathbf{z}_{N_k k}]$.

Substituting (18) and (16) into (19) and taking terms that are independent of $\mathbf{y}_k$ as a constant, the inner posterior $p(\mathbf{y}_k|\mathbf{X}_k)$ can be derived by completing the quadratic form of a Gaussian distribution, leading to

$$\mathbf{y}_k|\mathbf{X}_k \sim \mathcal{N}\Big(\mathbf{M}_y^{(k)}\mathbf{W}_y^{\top}\Psi\bar{\mathbf{x}}_k, \sigma^2\mathbf{M}_y^{(k)}\Big) \tag{20}$$

where $\bar{\mathbf{x}}_k = \sum_{j=1}^{N_k}\mathbf{x}_{jk}$, $\Psi = \mathbf{I} - \mathbf{W}_z\mathbf{M}_z\mathbf{W}_z^{\top}$ and $\mathbf{M}_y^{(k)} = (N_k\mathbf{W}_y^{\top}\Psi\mathbf{W}_y + \sigma^2\mathbf{I})^{-1}$.

With the above results, we can calculate the expectations $\langle\mathbf{f}_{jk}\rangle$ and $\langle\mathbf{f}_{jk}\mathbf{f}_{jk}^{\top}\rangle$ in terms of $p(\mathbf{f}_{jk}|\mathbf{X}_k)$ as follows:

$$\langle\mathbf{f}_{jk}\rangle = \Big[\langle\mathbf{y}_k\rangle^{\top}, \langle\mathbf{z}_{jk}\rangle^{\top}\Big]^{\top} \tag{21}$$

$$\langle \mathbf{f}_{jk}\mathbf{f}_{jk}^\top \rangle = \begin{bmatrix} \langle \mathbf{y}_k \mathbf{y}_k^\top \rangle & \langle \mathbf{y}_k \mathbf{z}_{jk}^\top \rangle \\ \langle \mathbf{z}_{jk} \mathbf{y}_k^\top \rangle & \langle \mathbf{z}_{jk} \mathbf{z}_{jk}^\top \rangle \end{bmatrix} \tag{22}$$

where

$$\begin{aligned} \langle \mathbf{y}_k \rangle &= \mathbb{E}\big[\mathbf{y}_k\big]_{\mathbf{y}_k|\mathbf{X}_k} = \mathbf{M}_y^{(k)} \mathbf{W}_y^\top \Psi \bar{\mathbf{x}}_k \\ \langle \mathbf{z}_{jk} \rangle &= \mathbb{E}\Big[\mathbb{E}\big[\mathbf{z}_{jk}\big]_{\mathbf{z}_{jk}|\mathbf{x}_{jk},\mathbf{y}_k}\Big]_{\mathbf{y}_k|\mathbf{X}_k} \\ &= \mathbf{M}_z \mathbf{W}_z^\top \big(\mathbf{x}_{jk} - \mathbf{W}_y \langle \mathbf{y}_k \rangle\big) \\ \langle \mathbf{y}_k \mathbf{y}_k^\top \rangle &= \sigma^2 \mathbf{M}_y^{(k)} + \langle \mathbf{y}_k \rangle \langle \mathbf{y}_k \rangle^\top \\ \langle \mathbf{z}_{jk} \mathbf{y}_k^\top \rangle &= \mathbf{M}_z \mathbf{W}_z^\top \Big(\mathbf{x}_{jk} \langle \mathbf{y}_k \rangle^\top - \mathbf{W}_y \langle \mathbf{y}_k \mathbf{y}_k^\top \rangle\Big) \\ \langle \mathbf{z}_{jk} \mathbf{z}_{jk}^\top \rangle &= \sigma^2 \mathbf{M}_z + \mathbf{M}_z \mathbf{W}_z^\top \mathbf{H}_{ij} \mathbf{W}_z \mathbf{M}_z \\ \mathbf{H}_{ij} &= \mathbf{x}_{jk} \mathbf{x}_{jk}^\top + \mathbf{W}_y \langle \mathbf{y}_k \mathbf{y}_k^\top \rangle \mathbf{W}_y^\top \\ &\quad - \mathbf{x}_{jk} \langle \mathbf{y}_k \rangle^\top \mathbf{W}_y^\top - \mathbf{W}_y \langle \mathbf{y}_k \rangle \mathbf{x}_{jk}^\top. \end{aligned}$$

2) CM-Step: In the CM-step, we alternately and conditionally estimate the column and row factor matrices by maximizing the log-likelihood function (17) with respect to $\mathbf{C}$ (or $\mathbf{R}$) with the other fixed.

With $\mathbf{R}$ fixed, the optimized $\mathbf{C}$ is given by

$$\tilde{\mathbf{C}} = \Bigg[\sum_{jk} \mathbf{X}_{jk} \mathbf{R} \ \mathrm{diag}\big(\langle \mathbf{f}_{jk} \rangle\big)\Bigg]\Bigg[\sum_{jk} \langle \mathbf{f}_{jk}\mathbf{f}_{jk}^\top \rangle \circledast \mathbf{R}^\top \mathbf{R}\Bigg]^{-1}. \tag{23}$$

After obtaining $\tilde{\mathbf{C}}$, $\mathbf{R}$ can be solved similarly as follows:

$$\tilde{\mathbf{R}} = \Bigg[\sum_{jk} \mathbf{X}_{jk}^\top \tilde{\mathbf{C}} \ \mathrm{diag}\big(\langle \mathbf{f}_{jk} \rangle\big)\Bigg]\Bigg[\sum_{jk} \langle \mathbf{f}_{jk}\mathbf{f}_{jk}^\top \rangle \circledast \tilde{\mathbf{C}}^\top \tilde{\mathbf{C}}\Bigg]^{-1}. \tag{24}$$

Finally, by maximizing (17) with respect to σ^2, the optimized noise variance can be obtained as follows:

$$\tilde{\sigma}^2 = \frac{1}{N d_c d_r} \sum_{jk} \Big\{ \mathrm{tr}\Big(\mathbf{X}_{jk}^\top \mathbf{X}_{jk}\Big) - \mathrm{tr}\Big(\mathbf{X}_{jk}^\top \tilde{\mathbf{C}} \ \mathrm{diag}\big(\langle \mathbf{f}_{jk} \rangle\big) \tilde{\mathbf{R}}^\top\Big)\Big\}. \tag{25}$$

3) Regularization on the CM-Step: Apart from utilizing the matrix structures and probabilistic modeling, we also introduce regularization in the CM-step for better generalization and more robustness against the SSS problem. Specifically, a multiple of the identity matrix $\gamma \mathbf{I}$ is added to the *second-order sample moment* $\sum_{jk} \langle \mathbf{f}_{jk}\mathbf{f}_{jk}^\top \rangle$ in (23) and (24), leading to the regularized updates for $\mathbf{C}$ and $\mathbf{R}$ as follows:

$$\tilde{\mathbf{C}} = \sum_{jk} \mathbf{X}_{jk} \mathbf{R} \ \mathrm{diag}\big(\langle \mathbf{f}_{jk} \rangle\big)\Big[\mathbf{F} \circledast \mathbf{R}^\top \mathbf{R}\Big]^{-1} \tag{26}$$

$$\tilde{\mathbf{R}} = \sum_{jk} \mathbf{X}_{jk}^\top \tilde{\mathbf{C}} \ \mathrm{diag}\big(\langle \mathbf{f}_{jk} \rangle\big)\Big[\mathbf{F} \circledast \tilde{\mathbf{C}}^\top \tilde{\mathbf{C}}\Big]^{-1} \tag{27}$$

where $\mathbf{F} = \gamma \mathbf{I} + \sum_{jk} \langle \mathbf{f}_{jk}\mathbf{f}_{jk}^\top \rangle$, and γ is the regularization parameter. The regularized terms stabilize the matrix inverses in (23) and (24), and consequently improve the generalization ability of PRODA.

By alternating between the E-step and CM-step until convergence, we can obtain the MLE solutions for $\boldsymbol{\theta}$. Algorithm 1 summarizes the ECM algorithm for PRODA. It can be proved that the ECM algorithm always increases the log-likelihood function (17) and is guaranteed to converge to local optima [39].

Algorithm 1 ECM Algorithm for PRODA

```
Input: The training set $\{\{\mathbf{X}_{jk}\}_{j=1}^{N_k}\}_{k=1}^{K}$, the number of class features $P_y$, the number of individual features $P_z$, and the regularization parameter $\gamma$.
Initialize $\mathbf{C}$, $\mathbf{R}$, and $\sigma^2$ randomly or in other ways.
Normalize each column of $\mathbf{C}$ and $\mathbf{R}$ to have unit norm.
repeat
    Compute the expectations $\langle \mathbf{f}_{jk} \rangle$ and $\langle \mathbf{f}_{jk}\mathbf{f}_{jk}^\top \rangle$ via (21) and (22), respectively.
    Update $\mathbf{C}$, $\mathbf{R}$, and $\sigma^2$ via (26), (27), and (25), respectively.
until convergence.
Output: $\mathbf{C}$, $\mathbf{R}$, and $\sigma^2$.
```

C. Initialization and Prediction

1) Initialization: The parameters $\boldsymbol{\theta} = \{\mathbf{C}, \mathbf{R}, \sigma^2\}$ of PRODA can be initialized randomly, so that unnecessary biases would not be introduced at the beginning of the iteration. Then, the columns of $\mathbf{C}$ and $\mathbf{R}$ are normalized to have unit length so that each initialized latent factor contributes equally to the PRODA model. Recall that $\mathbf{C} = [\mathbf{C}_y, \mathbf{C}_z]$ and $\mathbf{R} = [\mathbf{R}_y, \mathbf{R}_z]$ are constructed by the collective and individual factors. As the discriminative factor matrices, $\mathbf{C}_y$ and $\mathbf{R}_y$ are expected to project data of the same class close together while those of different classes apart. Therefore, it is also reasonable to initialize $\mathbf{C}_y$ and $\mathbf{R}_y$ by using the learned projections of nonprobabilistic MLDAs such as TR1DA or UMLDA. On the other hand, since $\mathbf{C}_z$ and $\mathbf{R}_z$ are responsible for capturing the structured noise, they should still be initialized randomly to maintain noninformative.

2) Prediction: With the trained PRODA model, we can obtain the low-dimensional features of a new coming observation $\mathbf{X}$ by computing the expectation of the collective latent variable $\mathbf{y}$ in terms of the posterior distribution $p(\mathbf{y}|\mathrm{vec}(\mathbf{X}))$. From (16), we can integrate out $\mathbf{z}$ and obtain

$$\mathrm{vec}(\mathbf{X})|\mathbf{y} \sim \mathcal{N}\Big(\mathbf{W}_y \mathbf{y}, \mathbf{W}_z \mathbf{W}_z^\top + \sigma^2 \mathbf{I}\Big). \tag{28}$$

Then the predictive posterior distribution can be readily derived as follows:

$$\mathbf{y}|\mathrm{vec}(\mathbf{X}) \sim \mathcal{N}\Big(\Sigma^{-1} \mathbf{W}_z^\top \Big(\mathbf{W}_z \mathbf{W}_z^\top + \sigma^2 \mathbf{I}\Big)^{-1} \mathrm{vec}(\mathbf{X}), \Sigma^{-1}\Big) \tag{29}$$

where $\Sigma = \mathbf{I} + \mathbf{W}_y^\top (\mathbf{W}_z \mathbf{W}_z^\top + \sigma^2 \mathbf{I})^{-1} \mathbf{W}_y$. Finally, the desired latent features of $\mathbf{X}$ is given by the expectation $\mathbb{E}[\mathbf{y}|\mathrm{vec}(\mathbf{X})] = \Sigma^{-1} \mathbf{W}_z^\top (\mathbf{W}_z \mathbf{W}_z^\top + \sigma^2 \mathbf{I})^{-1} \mathrm{vec}(\mathbf{X})$.

D. Time Complexity Analysis

Since both PLDA and PRODA are under the EM framework, they have comparable time complexity. For simplicity, let $D = d_c d_r$ be the number of input features, and $P = P_y = P_z$ be the number of extracted features. The E-step of PRODA

TABLE I
DETAILED DESCRIPTIONS OF PRODA AND THE COMPETING METHODS

Method	Full name	Tensor structures	Data uncertainty	Convergence	Reference
LDA	Linear Discriminant Analysis	×	×	✓	[7]
PCA+LDA	Fisherface	×	×	✓	[40]
RLDA	Regularized Linear Discriminant Analysis	×	×	✓	[33]–[35]
NLDA	Null Linear Discriminant Analysis	×	×	✓	[41]
ULDA	Uncorrelated Linear Discriminant Analysis	×	×	✓	[42]
MULDA	Maximum Uncertainty Linear Discriminant Analysis	×	×	✓	[43]
DATER	Discriminant Analysis with TEnsor Representation	✓	×	×	[20]
GTDA	General Tensor Discriminant Analysis	✓	×	✓	[22]
TR1DA	Tensor Rank-One Discriminant Analysis	✓	×	✓	[23]
UMLDA	Uncorrelated Multilinear Discriminant Analysis	✓	×	×	[21]
PLDA	Probabilistic Linear Discriminant Analysis	×	✓	✓	[24]
PRODA	Probabilistic Rank-One Discriminant Analysis	✓	✓	✓	Proposed

takes $O(\mathrm{NDP}^2)$ for computing the expectations $\langle \mathbf{f}_{jk} \rangle$ and $\langle \mathbf{f}_{jk}\mathbf{f}_{jk}^\top \rangle$. The M-step takes $O(\mathrm{NDP})$ for summing up the statistics, and $O(P^3)$ for matrix inverse. Therefore, the overall time complexity of PRODA is dominated by $O(\mathrm{TNDP}^2)$, where T is the number of iterations.

Please note that classical LDA takes $O(\mathrm{ND}^2)$ for computing the scatter matrices, and $O(D^3)$ for the generalized eigenvalue decomposition. Since $P < D$, each PRODA iteration is not much slower than LDA, provided that P is not too large. As will be shown in the next section, PRODA usually converges within a few iterations. Therefore, the overall computational cost of PRODA is acceptable for extracting a moderate number of features. For large scale problems, stochastic inference with mini-batch updates can be used to improve the efficiency of PRODA, which could be a future work.

IV. EXPERIMENTS

This section evaluates the performance of PRODA in supervised subspace learning, where a set of experiments are conducted to achieve the following two objectives.

1) Demonstrate the effectiveness of PRODA on classification tasks by comparing against LDA and its state-of-the-art extensions.
2) Investigate the behavior and properties of PRODA under different configurations.

A. Experimental Settings

PRODA is compared against LDA and its *linear*, *multilinear*, and *probabilistic* variants. Table I gives the detailed descriptions of PRODA and the competing methods.

1) Number of Extracted Features: For nonprobabilistic LDAs: LDA, PCA+LDA, RLDA, NLDA, ULDA, and MULDA, up to $K-1$ features are tested, which is the maximum number can be extracted. For MLDAs: DATER and GTDA, up to $30 \times 30 = 900$ features are tested. TR1DA is tested up to 500 features. UMLDA is tested up to 35 features by following the settings in [21]. For probabilistic methods: PLDA and PRODA, we set $P_y = P_z = 500$ for the collective and individual features, respectively. We have verified that extracting more features does not lead to better results with statistical significance for the competing methods.

2) Initialization: DATER, GTDA, TR1DA, UMLDA, PLDA, and PRODA are iterative methods and require initialization. DATER and GTDA are initialized by pseudo identity matrices. TR1DA and UMLDA are initialized uniformly [21], and PLDA is initialized randomly. These settings lead to the best results for the corresponding methods in our experiments. Unless otherwise specified, PRODA is initialized randomly.

3) Number of Iterations and Convergence Criteria: We set the number of iterations for DATER and GTDA to be 1, which results in the best performance in our experiments. TR1DA and UMLDA are tested with ten iterations by following the settings in [21]. For PLDA and PRODA, we iterate them until 300 iterations or the relative change of the log-likelihood is smaller than 10^{-4}.

4) Tuning Parameters: Originally, RLDA, TR1DA, UMLDA, and PRODA have tuning parameters to be determined. For fair comparison, the same regularization strategy of PRODA is also applied to PLDA, which introduces a regularization parameter for PLDA. We test these methods by selecting all the parameters from $\{10^{-5}, 10^{-4}, \dots, 10^{5}\}$, and report their best results.

B. Face Recognition

1) Datasets: Two face datasets are utilized. The first is the UMIST database [44], which consists of 575 images of 20 subjects. Images of each subject are taken in various poses from profile to frontal views with a neutral expression. The second one is the CMU PIE database [45]. It consists of 41 368 images of 68 subjects with four expressions, in 13 poses, and under 43 illumination conditions. We conduct the experiments on a subset of the CMU PIE database by selecting face images in seven poses (C05, C07, C09, C27, C29, C37, C11) and under 21 illumination conditions (02 to 22). This results in 9987 images in total, and around 147 samples per subject. All face images are normalized to 32×32 gray-level pixels.

2) Experimental Setup: Each dataset is randomly partitioned into training and test sets, leading to L images *per subject* for training and the rest for test. We extract low-dimensional features via each of the above-mentioned method, and then sort the extracted features in descending order by their Fisher scores [46]. After feature extraction, we follow the same settings of the competing methods [20]–[24] to use the

TABLE II
RECOGNITION RATES (MEAN±STD.%) ON THE UMIST DATASET (BEST, SECOND BEST, AND * STATISTICALLY COMPARABLE)

L	2	3	4	5	6	7	10	15
LDA	67.18±2.54	74.91±3.06	80.46±2.94	84.84±3.38	88.88±3.66	91.56±2.44	95.31±1.93	97.16±0.92
PCA+LDA	70.75±2.54	78.45±3.62	82.12±7.80	84.88±4.17	90.90±5.33	88.44±5.83	96.72±1.42	98.40±0.79
RLDA	74.97±3.05	86.56±4.10*	90.99± 2.66	93.12±2.50*	96.02±2.34	97.95±1.12*	99.01±0.58	99.82±0.35*
NLDA	74.97±3.05	85.53±3.20	90.36±2.71	92.00±2.76	95.76±2.49	97.01±0.71	98.56±0.87	99.27±0.42
ULDA	66.86±2.18	74.95±2.32	80.10±3.04	84.55±3.21	88.20±3.46	91.47±2.20	94.75±1.78	97.31±0.88
MULDA	70.79±2.63	83.48±3.49	89.09±2.18	91.87±3.62	95.38±2.13	97.79±0.99*	98.91±0.68	99.75±0.39*
DATER	77.46±4.61	87.55±2.58*	91.58±1.90	93.24±2.66*	96.44±1.81*	97.86±1.05	98.88±0.45	99.78±0.31*
GTDA	77.98±5.04	86.23±3.09	91.39±1.60	93.22±2.94*	95.91±1.96	97.82±0.80*	98.59±0.50	99.24±0.58
TR1DA	72.24±5.36	82.33±2.37	86.38±2.94	89.43±3.27	93.89±3.45	96.09±1.51	98.40±0.74	99.42±0.52
UMLDA	**81.85±2.98***	86.87±2.91	92.40 ± 2.74*	**94.44±2.17***	**97.16±1.85***	98.00±0.60*	98.96±0.54	99.60±0.36
PLDA	77.33±3.17	85.32±2.54	90.75± 2.24	93.03 ± 2.87*	96.31±2.41*	97.95±0.91*	99.28±0.33*	99.67±0.36*
PRODA	80.56±3.66*	**88.35±2.32***	**93.29±2.09***	94.36±2.44*	96.90±1.85*	**98.48±0.69***	**99.41±0.37***	**99.93±0.15***

TABLE III
RECOGNITION RATES (MEAN±STD.%) ON THE CMU PIE DATASET (BEST, SECOND BEST, AND * STATISTICALLY COMPARABLE)

L	2	3	4	5	6	7	10	40
LDA	43.73±1.75	55.33±1.28	62.82±1.69	67.70±0.99	70.49±0.88	73.07±0.63	75.17±1.14	98.55±0.21
PCA+LDA	34.55±1.82	47.94±1.95	59.19±2.48	66.40±1.67	70.95±0.74	74.80±0.93	79.06±1.26	98.63±0.18
RLDA	44.47±1.93	58.56 ±1.95	67.36±2.21	73.72±1.69	78.02±1.02	82.51±0.84	87.74±0.94	99.16±0.18
NLDA	44.47±1.94	57.76±1.96	65.84±2.46	71.43±1.53	74.47±0.81	77.74±1.08	80.88±1.25	-
ULDA	43.96±1.66	55.55±1.37	62.89±1.66	67.84±1.07	70.74±0.78	73.14±0.65	75.08±1.16	97.69±0.28
MULDA	36.90±1.67	50.21±1.35	58.77±1.62	66.90±0.72	72.07±0.83	77.10±0.85	85.59±0.67	98.82±0.19
DATER	40.41±4.06	56.14±1.73	63.62±1.46	68.88±0.98	72.62±1.42	75.86±1.61	82.52±1.17	97.14±0.39
GTDA	41.15±3.18	52.01±2.09	56.85±2.52	61.90±0.84	65.91±1.55	69.68±1.84	76.68±0.99	96.72±0.50
TR1DA	34.19±2.42	42.22±1.53	48.28±1.87	53.96±1.52	57.11±1.32	61.25±1.07	67.63±1.77	92.16±0.70
UMLDA	39.91±1.77	50.83±1.72	58.42±1.65	63.31±1.40	66.95±1.10	70.86±0.73	76.80±1.30	94.27±0.21
PLDA	44.69 ± 1.81	58.10±1.91	66.98±2.28	73.42 ± 1.46	77.64±0.95	81.85±0.93	**88.11±0.89***	**99.27±0.17***
PRODA	**47.86±2.22***	**62.78±1.72***	**70.63±2.22***	**76.06±1.39***	**79.38±1.07***	**82.98±1.16***	87.55±0.60*	98.67±0.25

nearest neighbor classifier for classification, which is trained with different numbers of the extracted features (up to the maximums). We conduct experiments over ten such random partitions, and report the best average recognition rates with the standard deviations for each method. The best results are highlighted in bold font, and the second best ones are *underlined*. We also mark the comparable results with an asterisk * based on a t-test with a p-value of 0.06.

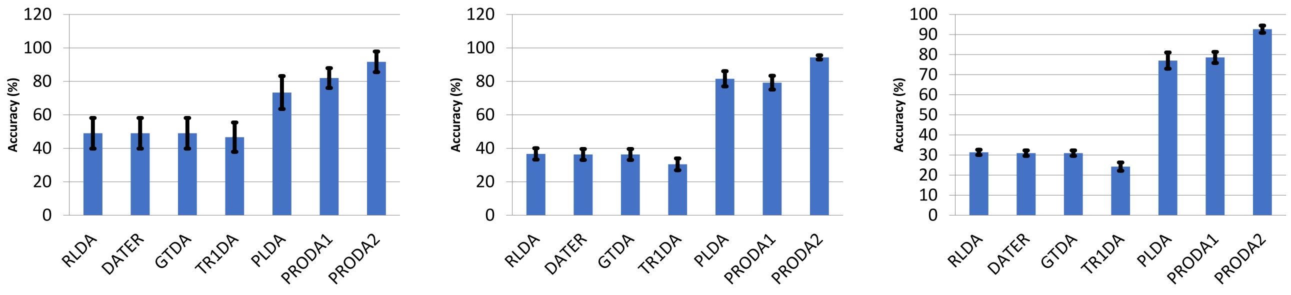
3) Result Analysis: Table II shows the recognition rates on the UMIST dataset. Most *nonprobabilistic linear* extensions of LDA get significantly better results than the baseline, among which RLDA is the best method. On average, all MLDAs except TR1DA achieve comparable or even better performance than RLDA, which indicates that utilizing the matrix structures can improve the performance of discriminant analysis.

UMLDA and PRODA are the best two methods on the UMIST dataset, which achieve comparably good performance except for $L = 3$. This could be attributed to both the exploited matrix structures and their individual properties preserved in the extracted features. Specifically, UMLDA imposes uncorrelated constraints to extract independent features, while PRODA learns a flexible generative model to capture generic data characteristics.

Table III shows the recognition rates on the CMU PIE dataset. PRODA achieves the best performance with statistical significance in most cases. Specifically, it outperforms the second best results 2.47% on average with $L = 2 \sim 7$, and is more advantageous with small training sizes ($L = 2, 3, 4$). This implies that PRODA is more robust than the competing methods against the SSS problem. Although PRODA is worse than PLDA by 0.58% on average with $L = 10, 40$, it still achieves reasonably a good performance, and consistently outperforms other MLDAs. It is worth noting that NLDA does not work on the CMU PIE dataset with $L = 40$, since the null space of the within-class scatter matrix becomes noninformative and only contains zero vectors when the training size is large.

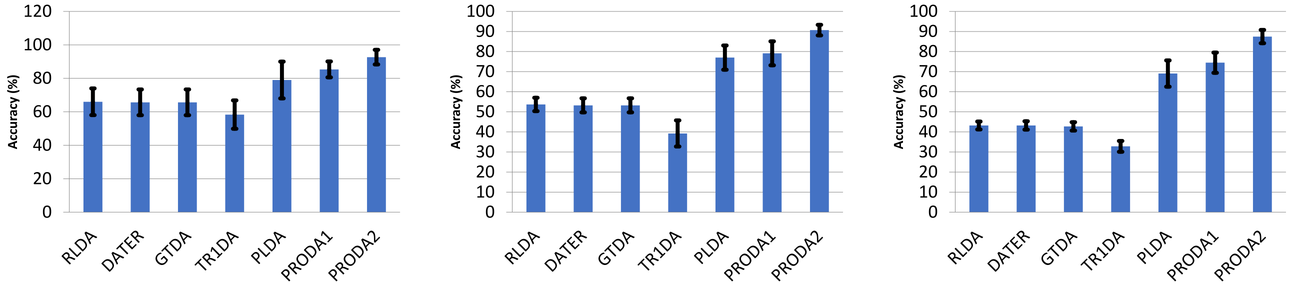
Unlike the experiments on the UMIST dataset, UMLDA fails to perform well on the CMU PIE dataset. Such difference of the UMLDA performance could be attributed to the uncorrelated constraints imposed by UMLDA and the different characteristics of the two datasets. The UMIST dataset consists of face images covering a range of poses from 20 subjects (classes), where the samples of each class have strong correlations. On the other hand, the CMU PIE dataset is more challenging, where face images are from 68 subjects and under both pose and illumination variations. Due to the uncorrelated constraints, UMLDA is effective in extracting the discriminative features from the UMIST dataset. However, when it comes to the CMU PIE dataset, the uncorrelated constraints could be too restricted to characterize both pose and illumination variations, leading to relatively poor results.

For the both datasets, *ratio*-based MLDAs such as DATER and UMLDA outperform the *difference*-based ones such as



(a) (b) (c)

Fig. 2. Sketch recognition rates with T training photographs (PvS) on the CUHK student dataset. (a) $T = 30$. (b) $T = 90$. (c) $T = 150$.



(a) (b) (c)

Fig. 3. photograph recognition rates with T training sketches (SvP) on the CUHK student dataset. (a) $T = 30$. (b) $T = 90$. (c) $T = 150$.

GTDA and TR1DA in most cases. This indicates that the good convergence property of GTDA and TR1DA may come at the expense of their feature discriminability. On the other hand, although DATER and UMLDA perform better than their difference-based counterparts, they have no convergence guarantee, and may test different iteration numbers to find the best results. In contrast, PRODA overcomes the limitations of both the ratio- and difference-based MLDAs by learning matrix subspaces under the probabilistic framework, leading to the best performance and the convergence guarantee.

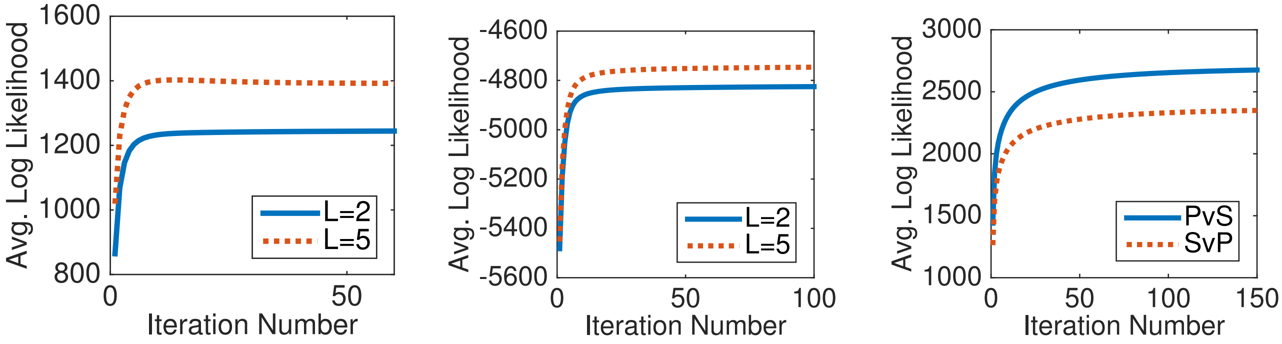
To verify whether the probabilistic framework itself helps classification, we develop a degenerated version of PRODA (denoted by PRODA*), where the posterior expectation (22) is replaced by its point estimation during the ECM updates. To be more specific, $\langle \mathbf{y}_k \mathbf{y}_k{}^\top \rangle$ and $\langle \mathbf{z}_{jk} \mathbf{z}_{jk}^\top \rangle$ are replaced by $\langle \mathbf{y}_k \rangle \langle \mathbf{y}_k \rangle^\top$ and $\langle \mathbf{z}_{jk} \rangle \langle \mathbf{z}_{jk} \rangle^\top$, respectively. Due to this modification, the E-step of PRODA* ignores some uncertainty information captured by the probabilistic framework, i.e., the covariance matrices $\sigma^2 \mathbf{M}_y^{(k)}$ and $\sigma^2 \mathbf{M}_z$. We test PRODA* on the UMIST and CMU PIE datasets. The experimental results show that PRODA* achieves almost the same performance with PRODA for the UMIST dataset, whereas it is worse than PRODA by 1.49% on average for the CMU PIE dataset. This indicates that by estimating the covariance matrices, the probabilistic framework can improve the classification performance.

C. Facial Photograph-Sketch Matching

1) Dataset: The Chinese University of Hong Kong (CUHK) student database [47] is tested, which consists of 188 subjects. Each subject has a facial photograph and a corresponding sketch of the photograph in the frontal pose, under the normal illumination condition, and with the neutral expression. Each image is resized to 40 × 32 gray-level pixels.

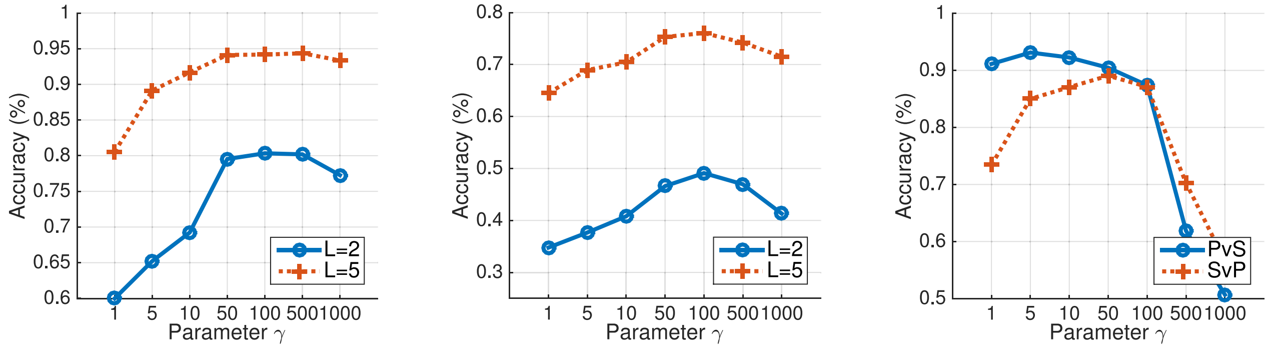
2) Experimental Setup: We randomly select $T = 30, 90, 150$ subjects from the CUHK student dataset. The task is to recognize the sketches (or photographs) of each subject by observing the others, leading to T photographs/sketches for training and the corresponding T sketches/photographs for test. For each T, we conduct such experiments ten times, and report the average recognition rates. This is both a single-sample and heterogeneous classification problem, making face recognition more challenging. As a consequence, LDA, PCA+LDA, NLDA, ULDA, MULDA, and UMLDA are inapplicable due to the badly ill-conditioned within-class scatter matrix. We test both the random and TR1DA-based schemes in Section III-C for initializing PRODA, which are indicated as PRODA1 and PRODA2, respectively.

3) Result Analysis: Figs. 2 and 3 show the recognition rates on the CUHK student dataset for the training photograph versus testing sketch (PvS) and training sketch versus testing photograph (SvP) cases, respectively. As can be seen, nonprobabilistic LDAs fail to perform well, while the probabilistic ones obtain significantly better results. Since each subject only has a single training sample, it is almost impossible to estimate the between- and within-class scatter accurately. This is probably responsible for the poor results of RLDA, DATER, GTDA, and TR1DA, since their performance is highly dependent on the quality of the scatter matrices. On the other hand, PLDA and PRODA are more robust against the so-called "single sample per person" problem, because they implicitly model the



(a) (b) (c)

Fig. 4. Average log-likelihood of PRODA at each iteration. (a) UMIST. (b) CMU PIE. (c) CUHK with $T = 150$.



(a) (b) (c)

Fig. 5. Recognition rates of PRODA in different settings of the regularization parameter γ. (a) UMIST. (b) CMU PIE. (c) CUHK with $T = 150$.

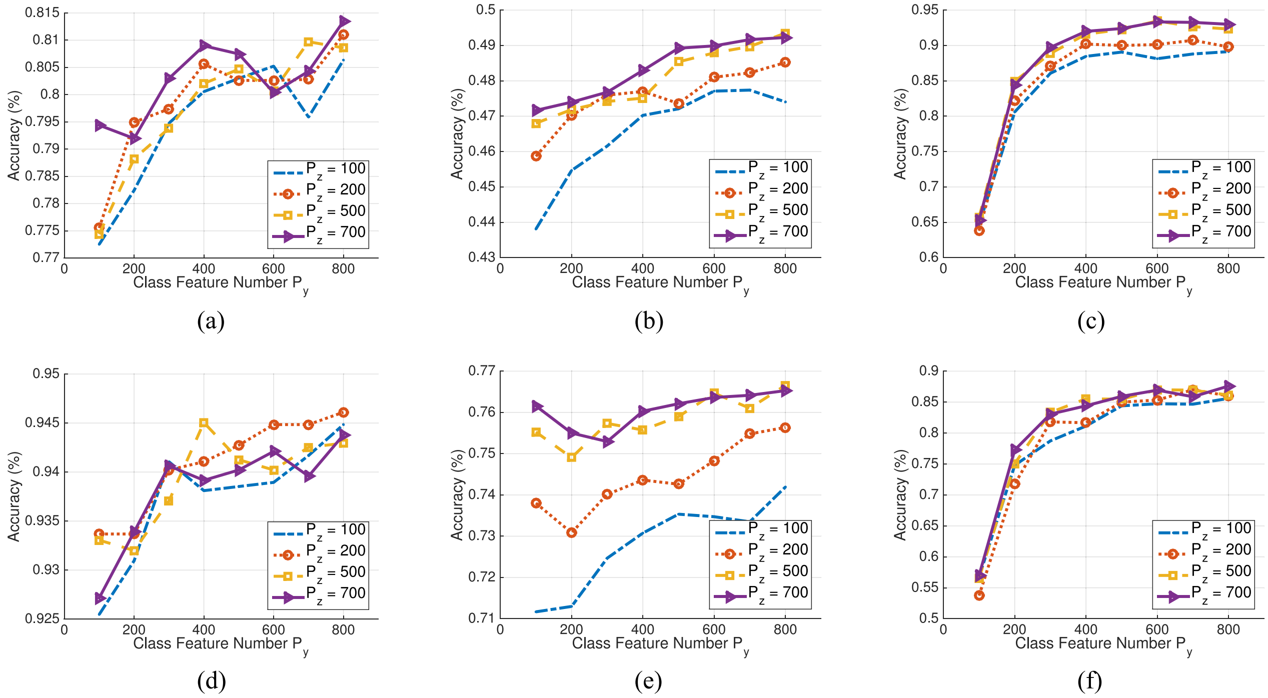
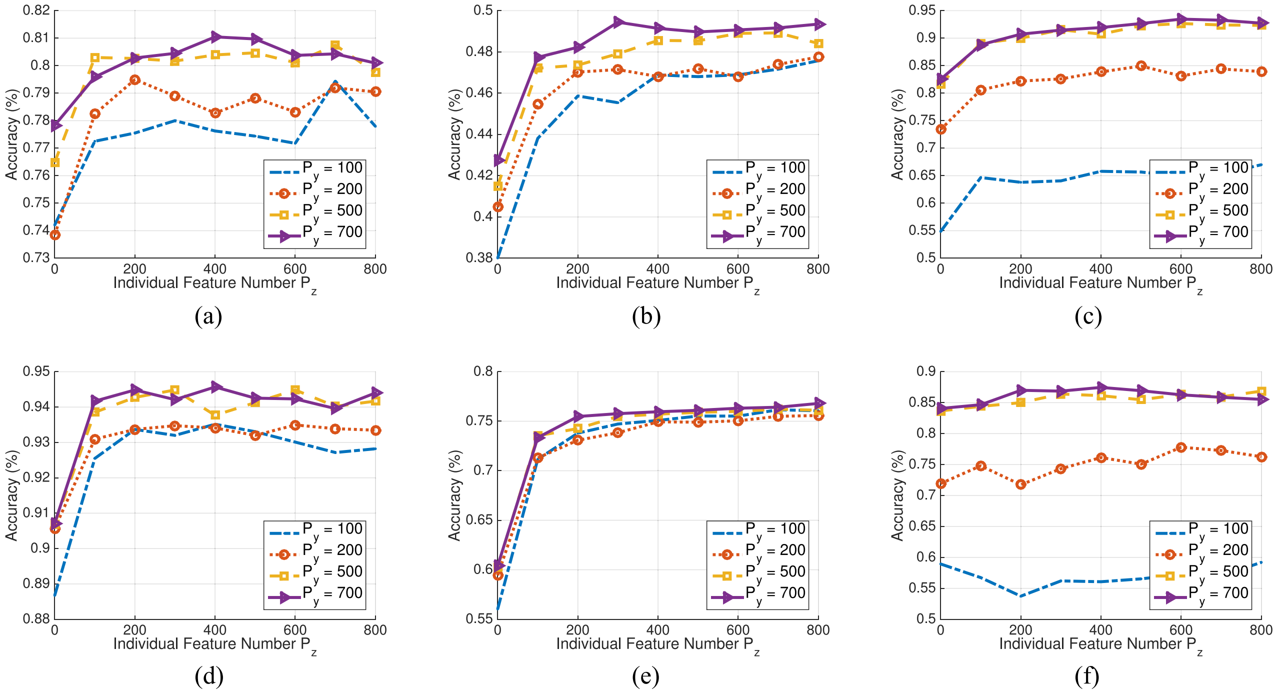


Fig. 6. Recognition rates of PRODA as P_y increases with P_z fixed. (a) UMIST with $L = 2$. (b) CMU PIE with $L = 2$. (c) CUHK PvS with $T = 150$. (d) UMIST with $L = 5$. (e) CMU PIE with $L = 5$. (f) CUHK SvP with $T = 150$.

between- and within-class information under the probabilistic framework rather than explicitly manipulate the inaccurate scatter matrices.

Although both PLDA and PRODA take advantage of the probabilistic framework, PRODA consistently outperforms PLDA except the PvS case with $T = 90$. This demonstrates



(a) (b) (c)

(d) (e) (f)

Fig. 7. Recognition rates of PRODA as P_z increases with P_y fixed. (a) UMIST with $L = 2$. (b) CMU PIE with $L = 2$. (c) CUHK PvS with $T = 150$. (d) UMIST with $L = 5$. (e) CMU PIE with $L = 5$. (f) CUHK SvP with $T = 150$.

the effectiveness of the matrix structures in improving the performance of subspace learning. In addition, since PRODA has less model complexity with fewer parameters, the estimation of PRODA with limited sample sizes should be more robust than that of PLDA. This may also contribute to the good performance of PRODA on the CUHK student dataset. It is worth noting that although TR1DA only obtains poor results, it seems to be a good initialization for PRODA, which leads to great improvements over the randomly initialized PRODA. This implies that data-dependent initializations based on other MLDAs can help PRODA to escape bad local optima, which would be more preferable than the random scheme especially when there is only a small number of training samples.

D. Convergence and Parameter Sensitivity

This section empirically studies the convergence property and the parameter sensitivity of PRODA.

1) Convergence: The convergence property of PRODA is first tested with $P_y = P_z = 500$, where we set $\gamma = 100$ for the UMIST and CMU PIE datasets, and $\gamma = 10$ for the CUHK student dataset. Fig. 4 shows the average log-likelihood of PRODA at each iteration on the UMIST, CMU PIE, and CUHK student datasets, respectively. PRODA monotonically increases the log-likelihood and converges in a few iterations, which supports its theoretical convergence guarantee.

2) Regularization Parameter γ: We then study how the performance of PRODA changes with the regularization parameter γ by fixing $P_y = P_z = 500$. As can be seen in Fig. 5, the imposed regularization in the CM-step effectively improves the performance of PRODA. In addition, the best choice of γ seems to be around 50–100 for all the involved datasets in different settings. This suggests that the best γ is not very sensitive to different training settings.

3) Collective and Individual Feature Numbers P_y and P_z: Finally, with the same γ settings in the convergence study, the effects of P_y and P_z are investigated. Fig. 6 shows the recognition rates of PRODA as P_y increases with P_z fixed. With a fixed P_z, the recognition rates of PRODA become higher as P_y increases, which suggests that a relatively large P_y is desirable for PRODA. This is expectable because there are more and more features available for capturing the discriminative information as P_y increases.

Fig. 7 shows the recognition rates of PRODA as P_z increases with P_y fixed. PRODA does not perform well when $P_z = 0$. This indicates that it is necessary and important to learn both the collective and individual subspaces for extracting more discriminative features. Increasing P_z improves the performance of PRODA at the beginning, while it no longer leads to better results after $P_z > 300$. This suggests that a medium P_z is enough for PRODA to get good results.

V. Conclusion

We have proposed PRODA for learning discriminative subspaces from matrices. By representing each observation as a linear combination of collective and individual rank-one matrices, PRODA achieves the following desirable properties.

1) It is flexible in capturing discriminative features and nondiscriminant noise.
2) It exploits the matrix structures to obtain compact subspace representations, reduced model complexity, and robustness against the SSS problem.
3) It is guaranteed to converge to local optima without introducing additional tuning parameters.

These properties give PRODA the edge over both probabilistic and MLDA extensions. Experimental results on three real-world datasets have shown the superiority of PRODA to the competing methods.

Beyond PRODA, there are several sensible future research directions. For example, PRODA can be extended to nonlinear versions by mixing a set of PRODA models or employing Gaussian processes. In some applications, it is useful to impose uncorrelated or sparse constraints on subspace bases for better interpretation and less feature redundancy. This may motivate uncorrelated or sparse extensions of PRODA, which could be developed by introducing certain priors and using Bayesian nonparametric techniques. In addition, semi-supervised extensions of PRODA by making use of unlabeled data as in [48] could also be an interesting future work.

References

[1] J. Lu and Y.-P. Tan, "Regularized locality preserving projections and its extensions for face recognition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 958–963, Jun. 2010.

[2] Y.-M. Cheung, "*k**-means: A new generalized *k*-means clustering algorithm," *Pattern Recognit. Lett.*, vol. 24, no. 15, pp. 2883–2893, 2003.

[3] Y. Zhou and S. Sun, "Manifold partition discriminant analysis," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 830–840, Apr. 2017.

[4] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the l2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.

[5] Z.-L. Sun, D.-S. Huang, Y.-M. Cheung, J. Liu, and G.-B. Huang, "Using FCMC, FVS, and PCA techniques for feature extraction of multispectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 2, pp. 108–112, Apr. 2005.

[6] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer, 2002.

[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA, USA: Academic, 1990.

[8] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[9] Q. Shi, H. Lu, and Y. Cheung, "Rank-one matrix completion with automatic rank estimation via L1-Norm regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4744–4757, Oct. 2018, doi: 10.1109/TNNLS.2017.2766160.

[10] Q. Shi, H. Lu, and Y.-M. Cheung, "Tensor rank estimation and completion via CP-based nuclear norm," in *Proc. 26th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2017, pp. 949–958.

[11] Q. Shi, Y.-M. Cheung, and Q. Zhao, "Feature extraction for incomplete data via low-rank tucker decomposition," in *Proc. Eur. Conf. Mach. Learn. Principles Pract. Knowl. Disc. Databases (ECML-PKDD)*, 2017, pp. 564–581.

[12] A. Sharma and K. K. Paliwal, "Linear discriminant analysis for the small sample size problem: An overview," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 3, pp. 443–454, 2015.

[13] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

[14] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.

[15] X. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 4, pp. 1170–1175, Aug. 2010.

[16] J. Zhao, P. L. H. Yu, and J. T. Kwok, "Bilinear probabilistic principal component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 492–503, Mar. 2012.

[17] J. Yu, Y. Rui, Y. Y. Tang, and D. Tao, "High-order distance-based multiview stochastic learning in image classification," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2431–2442, Dec. 2014.

[18] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. Boca Raton, FL, USA: CRC, 2013.

[19] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1569–1576.

[20] S. Yan *et al.*, "Discriminant analysis with tensor representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. San Diego, CA, USA, 2005, pp. 526–532.

[21] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 103–123, Jan. 2009.

[22] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[23] D. Tao, X. Li, X. Wu, and S. Maybank, "Tensor rank one discriminant analysis—A convergent method for discriminative multilinear subspace selection," *Neurocomputing*, vol. 71, nos. 10–12, pp. 1866–1882, 2008.

[24] S. Prince, P. Li, Y. Fu, U. Mohammed, and J. Elder, "Probabilistic models for inference about identity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 144–157, Jan. 2012.

[25] L. El Shafey, C. McCool, R. Wallace, and S. Marcel, "A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1788–1794, Jul. 2013.

[26] A. Sizov, K. A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *Proc. Joint IAPR Int. Workshops Struct. Syntactic Stat. Pattern Recognit.*, 2014, pp. 464–475.

[27] S. Ioffe, "Probabilistic linear discriminant analysis," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 531–542.

[28] S. Yu, K. Yu, V. Tresp, H.-P. Kriegel, and M. Wu, "Supervised probabilistic principal component analysis," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2006, pp. 464–473.

[29] Z. Li, D. Gong, Q. Li, D. Tao, and X. Li, "Mutual component analysis for heterogeneous face recognition," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 3, p. 28, 2016.

[30] Q. Li, W. Bian, R. Y. Da Xu, J. You, and D. Tao, "Random mixed field model for mixed-attribute data restoration," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1244–1250.

[31] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognit.*, vol. 44, no. 7, pp. 1540–1551, 2011.

[32] W. Cao *et al.*, "Total variation regularized tensor RPCA for background subtraction from compressive measurements," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4075–4090, Sep. 2016.

[33] J. H. Friedman, "Regularized discriminant analysis," *J. Amer. Stat. Assoc.*, vol. 84, no. 405, pp. 165–175, 1989.

[34] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Regularized discriminant analysis for the small sample size problem in face recognition," *Pattern Recognit. Lett.*, vol. 24, no. 16, pp. 3079–3087, 2003.

[35] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition," *Pattern Recognit. Lett.*, vol. 26, no. 2, pp. 181–191, 2005.

[36] K. Inoue and K. Urahama, "Non-iterative two-dimensional linear discriminant analysis," in *Proc. 18th Int. Conf. Pattern Recognit.*, vol. 2. Hong Kong, 2006, pp. 540–543.

[37] D. Xu *et al.*, "Human gait recognition with matrix representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 7, pp. 896–903, Jul. 2006.

[38] D. Luo, C. Ding, and H. Huang, "Symmetric two dimensional linear discriminant analysis (2DLDA)," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2820–2827.

[39] X.-L. Meng and D. B. Rubin, "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.

[40] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[41] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognit.*, vol. 33, no. 10, pp. 1713–1726, 2000.

[42] J. Ye, R. Janardan, Q. Li, and H. Park, "Feature extraction via generalized uncorrelated linear discriminant analysis," in *Proc. ACM 21st Int. Conf. Mach. Learn.*, 2004, pp. 895–902.

[43] T. Kasparek *et al.*, "Maximum-uncertainty linear discrimination analysis of first-episode schizophrenia subjects," *Psychiatry Res. Neuroimag.*, vol. 191, no. 3, pp. 174–181, 2011.

[44] D. B. Graham and N. M. Allinson, “Characterising virtual eigensignatures for general purpose face recognition,” in *Face Recognition*. Heidelberg, Germany: Springer, 1998, pp. 446–456.

[45] T. Sim, S. Baker, and M. Bsat, “The CMU pose, illumination, and expression database,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

[46] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2012.

[47] X. Tang and X. Wang, “Face sketch synthesis and recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 687–694.

[48] Y. Zhang and D.-Y. Yeung, “Heteroscedastic probabilistic linear discriminant analysis with semi-supervised extension,” in *Proc. Mach. Learn. Knowl. Disc. Databases*, 2009, pp. 602–616.

Yang Zhou received the M.Eng. degree in computer technology from the Department of Computer Science and Technology, East China Normal University, Shanghai, China, in 2015. He is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science, Hong Kong Baptist University, Hong Kong.

His current research interests include probabilistic modeling, tensor analysis, and machine learning.

Yiu-ming Cheung (F’18) received the Ph.D. degree in computer science and engineering from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

He is currently a Full Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include machine learning, pattern recognition, visual computing, and optimization.

Dr. Cheung serves as an Associate Editor for many journals, including IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and *Pattern Recognition*. He is an IET Fellow, a BCS Fellow, an RSA Fellow, and an IETI Distinguished Fellow. More details can be found at: http://www.comp.hkbu.edu.hk/∽ymc.