Neurocomputing 512 (2022) 100–116

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Orthogonal multi-view analysis by successive approximations via eigenvectors

Li Wang [a,b,*], Lei-Hong Zhang [c], Chungen Shen [d], Ren-Cang Li [b]

[a] Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019-0408, USA
[b] Department of Mathematics, University of Texas at Arlington, Arlington, TX 76019-0408, USA
[c] School of Mathematical Sciences and Institute of Computational Science, Soochow University, Suzhou 215006, Jiangsu, China
[d] College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China

ARTICLE INFO

Article history:
Received 24 September 2021
Revised 17 July 2022
Accepted 4 September 2022
Available online 8 September 2022
Communicated by Zidong Wang

Keywords:
Orthogonal multi-view analysis
successive approximations via eigenvectors
Krylov subspace method

ABSTRACT

Orthogonality has been demonstrated to admit many desirable properties such as noise-tolerant, good for data visualization, and preserving distances. However, it is often incompatible with existing models and the resulting optimization problem is challenging even if compatible. To address these issues, we propose a trace ratio formulation for multi-view subspace learning to learn individual orthogonal projections for all views. The proposed formulation integrates the correlations within multiple views, supervised discriminant capacity, and distance preservation in a concise and compact way. It not only includes several existing models as special cases, but also inspires new models. Moreover, an efficient numerical method based on successive approximations via eigenvectors is presented to solve the associated optimization problem. The method is built upon an iterative Krylov subspace method which can easily scale up for high-dimensional datasets. Extensive experiments are conducted on various real-world datasets for multi-view discriminant analysis and multi-view multi-label classification. The experimental results demonstrate that the proposed models are consistently competitive to and often better than the compared methods.

1. Introduction

Multi-view data are increasingly collected for a variety of applications in the real world. They usually contain complementary, redundant, and corroborative contents and so are more informative than single-view data when it comes to characterize objects of the real-world. It is rather natural for human beings to perceive the world through comprehensive information collected by multiple sensory organs, but it is an open question on how to endow machines with analogous cognitive capabilities to do the same. To take full advantage of multi-view data, multi-view learning has attracted increasing attention due to its wide applications such as dimensionality reduction [1], cross-view recognition [2,3], clustering [4,5], classification [6], and multi-label learning [7,8]. Many learning criteria have been explored to capture the relations among multiple views including subspace learning methods [9,10], tensor approaches [11,12] and the deep learning [13–15].

Although great progress has been made by existing multi-view learning methods, there are still challenges. One of the fundamental challenges is how to represent and summarize multi-view data in such a way that comprehensive information concealed in multi-view data can be properly exploited by learning models. The heterogeneity gap [16] among multiple views makes it difficulty to construct such representations since features extracted from different views with similar semantics may be located in completely different subspaces, e.g., text is often symbolic while audio and image are signals. Another challenge is to deal with multi-view data of small to medium sizes. Notice that deep learning models have recently achieved impressive performance for various multi-view learning tasks [13–15], but they typically require much larger data sets and their learning complexity is significantly higher than shallow models [17].

A significant research effort has been about addressing the challenges by seeking a common semantic subspace into which the heterogeneous features from different views are projected. Multi-view subspace learning, as the most popularly studied methodology for multi-view learning [18,19], aims to narrow the heterogeneity gap under the assumption that all views are generated from a common latent space via some unknown transformations in the first place. The most representative subspace learning model is the canonical correlation analysis (CCA) [20], which was

* Corresponding author.

https://doi.org/10.1016/j.neucom.2022.09.018
0925-2312/

originally proposed to learn two linear projections by maximizing the cross-correlation between two views in a common space. It has since been extended to more than two views [21], nonlinear projections via kernel trick [22] and deep representation [23], supervised learning [24–27], sparse learning [28,9], and multi-output learning such as multi-label classification [29] and multi-target regression [30]. Recently, orthogonality has also been successfully explored in multi-view subspace learning, including orthogonal CCA (OCCA) [30–33], orthogonal multiset CCA (OMCCA) [33,34], and multi-view partial least squares (PLS) [35]. However, most multi-view subspace learning methods stay clear from orthogonality constraints [24,25] due to issues including model incompatibility (adding orthogonality constraints may cause incompatibility to inherent constraints already there in existing models) and optimization difficulty (even if there is no incompatibility issue, the resulting optimization problem is generally hard to solve under orthogonality constraints), let alone integrate with other learning criteria such as supervised information.

To address the above issues, we propose a trace ratio formulation for multi-view analysis with orthogonality constraints and an efficient method to solve the resulting optimization problem. To resolve the model incompatibility issue, we take the trace ratio formulation to model the pairwise correlations of multiple views by strictly following their original definitions. With the trace ratio formulation, orthogonality constraints are added without causing any incompatibility issue. Moreover, supervised information can be incorporated into the numerators or denominators of the trace ratios in order to capture the class separability or coherence. Although the trace ratio formulation is flexible, the resulting optimization problem is a challenging one. To solve the challenging problem, we propose an efficient optimization method called orthogonal successive approximation via eigenvectors (OSAVE).

Contributions. The main contributions of this paper are summarized as follows:

- We propose a trace ratio formulation for multi-view subspace learning, which can naturally integrate the dependency among multiple views, supervised information, and simultaneously learn orthogonal projections in a concise and compact form. We show that orthogonal linear discriminant analysis (OLDA), OCCA and OMCCA are special cases of the proposed formulation.
- Our formulation can be flexibly adapted for various learning scenarios. To justify the flexibility, we instantiate several new models from the proposed formulation. Three models are proposed for multi-view feature extraction, and two models for multi-view multi-label classification. Different from existing ones, our models are directly built on the essential trace ratio formulation with orthogonality constraints.
- To solve the challenging optimization problem of the proposed formulation, we present a successive approximation algorithm, which is built upon well-developed numerical linear algebra techniques. We describe an iterative Krylov subspace method for calculating the top eigenvector of generalized eigenvalue problem $A\boldsymbol{x} = \lambda B\boldsymbol{x}$ with possibly a singular B. The Krylov subspace method can serve as the workhorse for scalability.
- Extensive experiments are conducted for evaluating the proposed models against existing learning methods in terms of two learning tasks: multi-view feature extraction and multi-view multi-label classification. Experimental results on various real-world datasets demonstrate that our proposed models perform competitively to and often better than baselines.

Paper organization. We first present the scope of this paper in Section 2 and briefly review the relevant existing models in Section 3. In Section 4, we propose the novel trace ratio formulation for orthogonal multi-view analysis, and their instantiated models for multi-view discriminant analysis and multi-view multi-label classification. The proposed successive approximation algorithm is presented in Section 5 with its key component in A. Extensive experiments are conducted in Section 6. Finally, we draw our conclusions in Section 7.

2. Problem Setup and Necessary Statistics

Feature extraction is an important tool for multivariate data analysis. As multiple inputs may come from different sources (views), they are most likely heterogeneous and have large discrepancy among views. The aim of multi-view feature extraction is to exploit consensual, complementary, and overlapping information among views. In what follows, we first present the scope of this paper and then formulate some important statistics used through this paper.

2.1. Problem Description

Let $\left\{\left(\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(v)}, \boldsymbol{y}_i\right)\right\}_{i=1}^{n}$ be a dataset of v views, where the ith data points $\boldsymbol{x}_i^{(s)} \in \mathbb{R}^{d_s}$ of all views $(1 \leqslant s \leqslant v)$ are assumed to share the same class labels in $\boldsymbol{y}_i$ of c labels. The class labels can have different interpretations, dependent of the underlying learning task. For multi-output regression, $\boldsymbol{y}_i \in \mathbb{R}^c$, and it reduces to a scalar for the classical regression for which $c = 1$. For multi-label classification, $\boldsymbol{y}_i \in \{0,1\}^c$ with an understanding that the ith data points of all views have the class label r if $(\boldsymbol{y}_i)_r = 1$ and otherwise 0, where $(\boldsymbol{y}_i)_r$ is the rth entry of $\boldsymbol{y}_i$. If $\boldsymbol{1}_c^{\mathrm{T}}\boldsymbol{y}_i = 1$, then multi-label classification becomes a problem of c-class classification since one and only one class label is assigned to each instance of data points of all views. In particular, if $c = 2$ and $\boldsymbol{1}_c^{\mathrm{T}}\boldsymbol{y}_i = 1$, then it is just the binary classification.

In this paper, objective fulfilling linear transformations are sought to extract the latent representation for each view. Compared to deep neural networks, the studied shallow model can effectively work with multi-view data of small to medium sizes and orthogonality constraints. Therefore, this paper mainly concentrates on small- to medium-sized multi-view data. Let $P_s \in \mathbb{R}^{d_s \times k}$ be the projection matrix for view s to transform $\boldsymbol{x}_i^{(s)}$ from $\mathbb{R}^{d_s}$ to $\boldsymbol{z}_i^{(s)} = P_s^{\mathrm{T}}\boldsymbol{x}_i^{(s)}$ in the common space $\mathbb{R}^k$. Represent the n data points of view s by $X_s = \left[\boldsymbol{x}_1^{(s)}, \ldots, \boldsymbol{x}_n^{(s)}\right] \in \mathbb{R}^{d_s \times n}$ and its latent representation by $Z_s = \left[\boldsymbol{z}_1^{(s)}, \ldots, \boldsymbol{z}_n^{(s)}\right] = P_s^{\mathrm{T}}X_s \in \mathbb{R}^{k \times n}$. The goal of this paper is to learn projections $\{P_s\}$ from multi-view data with class labels. In addition, we focus on two learning tasks, i.e., multi-view feature extraction and multi-view multi-label classification, and using the latent representations learned from multi-view data to improve single-view classification performance. We first use a multi-view subspace learning method as a supervised dimensionality reduction step so that the embeddings obtained by the method hopefully encode important correlations among multiple views and their output labels, and then a base classification model is evaluated in the common space, e.g., one nearest neighbor classifier for multi-class classification [33] and multi-label k-nearest neighbor (ML-kNN) in the common space as the backend multi-label classifier [36]. It is expected to have better performance for the multi-view approach than any single-view method applied to each view only or to the naive concatenation approach in terms of both multiclass classification and multi-label classification.

2.2. Necessary Statistics

We denote the centered matrix of view s and the label matrix by

$$\widehat{X}_s = X_s H_s, \tag{1}$$
$$Y = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n], \tag{2}$$

respectively, where $H_n = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}$. The sample cross-covariance between view s and view t is

$$C_{s,t} = \widehat{X}_s\widehat{X}_t^{\mathrm{T}} = X_s H_n X_t^{\mathrm{T}}. \tag{3}$$

In particular, $C_{s,s}$ is the covariance of view s.

For the c-class classification, i.e., $Y \in \{0,1\}^{c\times n}$ and $\mathbf{1}_c^{\mathrm{T}}\boldsymbol{y}_i = 1$, we have the following properties:

$$Y^{\mathrm{T}}\mathbf{1}_c = \mathbf{1}_n, \Sigma = YY^{\mathrm{T}} = \mathrm{diag}(n_1, \ldots, n_c), \tag{4}$$

a diagonal matrix with the rth diagonal entry $n_r = \sum_{i=1}^{n}(\boldsymbol{y}_i)_r$ being the number of data points in class r. Let $Q = Y^{\mathrm{T}}\Sigma^{-1}Y$. The between-class scatter matrix is

$$S_{\mathrm{b}}^{(s)} = X_s\left(Q - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}\right)X_s^{\mathrm{T}}, \tag{5}$$

and the within-class scatter matrix takes the form

$$S_{\mathrm{w}}^{(s)} = C_{s,s} - S_{\mathrm{b}}^{(s)} = X_s(I - Q)X_s^{\mathrm{T}}. \tag{6}$$

3. Related Work

In this section, we briefly review the representative multi-view subspace methods with linear projections from unsupervised and supervised multiclass classification, as well as multi-label classification.

3.1. Unsupervised Methods

PLS and CCA can be directly applied to two-view data ($v = 2$) simply by replacing Y and the projection matrix of Y with X_2 and P_2 of view 2, respectively. For $v > 2$, the multi-set CCA (MCCA) [21]

$$\max_{\{P_s\in\mathbb{R}^{d_s\times k}\}} \sum_{s=1}^{v}\sum_{t=1}^{v}\mathrm{tr}\left(P_s^{\mathrm{T}}C_{s,t}P_t\right) \tag{7a}$$
$$\text{s.t.} \sum_{s=1}^{v}P_s^{\mathrm{T}}C_{s,s}P_s = I_k, \tag{7b}$$

is the most popularly used, chiefly due to its analytic solution via the generalized eigen-decomposition that has been well studied [37,38]. Orthogonal multiset CCA (OMCCA)

$$\max_{\{P_s\in\mathbb{R}^{d_s\times k}\}} \sum_{s=1}^{v}\sum_{t=1}^{v}\frac{\mathrm{tr}\left(P_s^{\mathrm{T}}C_{s,t}P_t\right)}{\sqrt{\mathrm{tr}\left(P_s^{\mathrm{T}}C_{s,s}P_s\right)P_t^{\mathrm{T}}}\sqrt{\mathrm{tr}\left(P_t^{\mathrm{T}}C_{t,t}P_t\right)}} \tag{8a}$$
$$\text{s.t.}P_s^{\mathrm{T}}P_s = I_k, \forall s \tag{8b}$$

is proposed in [33]. Its special case $v = 2$ is the orthogonal CCA (OCCA) [30–32]. In [34], a variant of (8) was studied. The key in (7) and (8) is the use of pairwise cross-covariance matrices $\{C_{s,t}\}$ to capture the consensus among the v views. Recently, PLS is extended for $v > 2$ in [35], too, where the orthogonality constraints $P_s^{\mathrm{T}}P_s = I_k$ for all s are imposed. Different from CCA, the cross-regression for multi-view feature extraction (CRMvFE) [10] is built on a regression model to learn two sets of projection matrices $\{P_s \in \mathbb{R}^{d_s\times k}\}$ and $\{F_s \in \mathbb{R}^{k\times d_s}\}$ by solving the following optimization problem

$$\min_{\{P_s\},\{F_s\}}\sum_{s,t}\|X_s - F_s^{\mathrm{T}}P_t^{\mathrm{T}}X_t\|\|_F^2 + \gamma\sum_{s=1}^{v}\|F_s\|_F^2 \tag{9a}$$
$$\text{s.t.}\sum_{s=1}^{v}P_s^{\mathrm{T}}P_s = I_k, \tag{9b}$$

and the robust CRMvFE (RCRMvFE) is also presented by replacing square loss with the $\ell_{2,1}$ norm. Instead of learning linear transformation functions, the low-dimensional representations can be directly optimized such as the similarity-consensus regularized multi-view manifold learning [1] and the multi-view Laplacian least squares [39]. The graph regularized MCCA[40] seeks both orthogonal common low-dimensional representations and single-view projection matrices by accounting for graph-induced knowledge of common sources. In addition to OMCCA and PLS, the above-mentioned methods do not attempt to obtain orthonormal projection matrices.

3.2. Supervised Methods

For supervised learning with multiclass labels, the output label Y can be naturally considered as a view of input [29]. However, the special structure of label information is neglected. To compensate that negligence and to take full advantage of multiclass label data, sophisticated multi-view feature extraction methods have been proposed. In [24], generalized multi-view analysis (GMA) is formulated, by integrating LDA (or some variants of it) and CCA, as

$$\max_{\{P_s\}}\sum_{s=1}^{v}\mathrm{tr}\left(P_s^{\mathrm{T}}S_{\mathrm{b}}^{(s)}P_s\right) + \sum_{s=1}^{v}\sum_{t=1,t\neq s}^{v}\alpha_{s,t}\mathrm{tr}\left(P_s^{\mathrm{T}}C_{s,t}P_t\right) \tag{10a}$$
$$\text{s.t.}P_s^{\mathrm{T}}S_{\mathrm{w}}^{(s)}P_s = I_k, \forall s, \tag{10b}$$

where $\alpha_{s,t}$ is the weight for cross-covariance between view s and view t. Unfortunately, this is a difficult optimization problem whose KKT condition leads to a multi-parameter eigenvalue problem like (28) later for which there is no efficient numerical method for its solution. For that reason, authors in [24] proposed to solve, instead, a relaxed problem, the same objective but a constraint different from (10b):

$$\sum_{s=1}^{v}\eta_s P_s^{\mathrm{T}}S_{\mathrm{w}}^{(s)}P_s = I_k, \tag{11}$$

resulting in a generalized eigenvalue problem [38], where $\{\eta_s\}$ are parameters to balance v independent constraints. $S_{\mathrm{b}}^{(s)}$ and $S_{\mathrm{w}}^{(s)}$ can be the ones for the classical LDA. Multi-view uncorrelated linear discriminant analysis (MULDA) [26] was proposed to replace (10b) with the uncorrelated constraints and

$$\sum_{s=1}^{v}\eta_s P_s^{\mathrm{T}}C_{s,s}P_s = I_k. \tag{12}$$

Multi-view modular discriminant analysis (MvMDA) [25] aims to maximize the distances between different class centers across different views and minimize the within-class scatter

$$\max_{\{P_s\}}\sum_{s=1}^{v}\sum_{t=1}^{v}\mathrm{tr}\left(P_s^{\mathrm{T}}X_s A X_t^{\mathrm{T}}P_t\right) : \text{s.t.}\sum_{s=1}^{v}P_s^{\mathrm{T}}S_{\mathrm{w}}^{(s)}P_s = I_k, \tag{13}$$

where $A = Y^{\mathrm{T}}\Sigma^{-1}H_c\Sigma^{-1}Y$. In addition, other learning criteria have been explored for the integrated model of supervised information and CCA. The sparse additive discriminative CCA (SaDCCA) [9] integrates supervised information into CCA with local diffusion process

to reflect the high-order characteristics of intra-class and the separability of inter-class as well as the sparsity of the projection matrices. SaDCCA is formulated as

$$\max_{\{P_s\}} \sum_{s=1}^{v} \mathrm{tr}\left(P_s^{\mathrm{T}} \widetilde{S}_{\mathrm{b}}^{(s)} P_s\right) + \sum_{s \neq t} \mathrm{tr}\left(P_s^{\mathrm{T}} C_{s,t} P_t\right) \tag{14a}$$

$$\text{s.t.} \ \sum_{s=1}^{v} P_s^{\mathrm{T}} C_{s,s} P_s = I_k, \|P_s\|_1 \leqslant \varepsilon_s, \forall s, \tag{14b}$$

where $\widetilde{S}_{\mathrm{b}}^{(s)}$ is the local inter-class scatter matrix based on diffusion on the tensor product graph, and ε_s is a small constant to control the ℓ_1 norms of projection matrices to induce sparsity in the matrices. In [2], the cross-view semantic consistency in the sample space, instead of the feature space, is proposed. In [41], a fractional-order embedding is learned to suppress the increase of eigenvalues calculated from noisy data with a small number of samples of high dimensions. A regression-based approach has also been explored to incorporate supervised information for multi-view learning. In [42], the robust adaptive weighting multi-view classification algorithm (RAMC) using robust loss with view importance learning and nonnegative ϵ-dragging to the label matrix is used to reduce the impact of outliers and noise in multi-view data. RAMC is formulated as the following problem

$$\min_{P_s, M \geqslant 0, \alpha} \|\sum_{s=1}^{v} \alpha_v P_s^{\mathrm{T}} X_s - (Y + Y \odot M)\|_{2,1} + \lambda \sum_{s=1}^{v} \|P_s\|_F^2 \tag{15a}$$

$$\text{s.t.} \ \alpha \geqslant 0, \mathbf{1}^{\mathrm{T}} \alpha = 1, \tag{15b}$$

where $\odot$ is the Hadamard product operator, M is the ϵ-dragging matrix, and α is the vector of view importance. RAMC shows superior classification performance to the adaptive-weighting discriminative regression model [6].

It is worth noting that imposing orthogonality constraints has attracted much attention in multi-view feature extraction in unsupervised learning, but it is seldom explored in supervised learning.

3.3. Multi-label Learning

Multi-label classification [43] is a kind of classification where one instance may have multiple labels from a set of predefined categories, i.e., a subset of labels. Because of multiple views of every instance, the multi-view multi-label classification data consist of multiple views. The situation is different from multi-view feature extraction, where each instance only has a single label. As there are a plenty of single-view multi-label methods in the literature [44,45], we will not review them all but apply ML-kNN [36] as the backbone of our multi-label classifier for multi-view multi-label classification since its good performance has been well verified on various single-view multi-label data sets.

Recently, a few multi-view multi-label classification methods have been proposed based on different learning criteria, including matrix factorization methods [46–48], CCA-based methods [49–51], a tensor-based method [52], a probabilistic model [7] and the deep learning method [8]. Note that matrix factorization methods often perform classification in the transductive semi-supervised manner, so it is not easy to be applied to unseen data. Among the other methods, CCA-based methods are most relevant to our proposed models. The methods [49,50] only work for cross-modal retrieval from one view to the other with provided labeled data as the supervised information, so they cannot be used for more than two views. The supervised multi-view multi-label canonical correlation projection (sM2CP) [51] extends the work in [49] for multi-view multi-label classification with more than two views. Specifically, sM2CP solves the following problem to obtain linear transformation matrices $\{P_s\}$:

$$\max_{\{P_s\}} \sum_{s \neq t} \mathrm{tr}\left(P_s^{\mathrm{T}} A_{s,t} P_s\right) : \text{s.t.} P_s^{\mathrm{T}} C_{s,s} P_s = I_k, \tag{16}$$

where $A_{s,t} = X_s^{\mathrm{T}} A^{\mathrm{multi}} X_t$ is the covariance matrix encoded with the multi-label class information through the cosine similarity A^{multi} whose (i,j)th entry is $\frac{\boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{y}_j}{\|\boldsymbol{y}_i\| \|\boldsymbol{y}_j\|}$. Although some other settings have been explored for multi-view multi-label data such as missing or incomplete data [53,54] and feature selection [55], they are different from the focus of this paper.

4. Orthogonal Multi-view Analysis

We propose a novel trace ratio formulation for multi-view discriminant analysis in order to learn orthogonal projections onto a latent common space.

4.1. Orthogonality and its Challenges

Researches have previously demonstrated that orthogonality built into single-view subspace learning models possesses desirable advantages such as more noise-tolerant, better suited for data visualization and distance preservation [56–61]. Among these advantages, distance preservation is one of the most important learning criteria, which has been successfully demonstrated in learning methods such as kernel learning [62] and density estimation [63,64].An orthogonal projection is able to preserve the pairwise distance if the vectors to be projected live in the range of the projection. Specifically, if $\boldsymbol{x}_i^{(s)} \in \mathscr{R}(P_s)$, the column subspace spanned by the columns of P_s, for all i and $P_s^{\mathrm{T}} P_s = I_k$, then we have $\boldsymbol{x}_i^{(s)} = P_s \widetilde{\boldsymbol{z}}_i^{(s)}$ for $\widetilde{\boldsymbol{z}}_i^{(s)} = P_s^{\mathrm{T}} \boldsymbol{x}_i^{(s)}$:

$$P_s \boldsymbol{z}_i^{(s)} = P_s \left(P_s^{\mathrm{T}} \boldsymbol{x}_i^{(s)}\right) = P_s \underbrace{P_s^{\mathrm{T}} P_s}_{} \widetilde{\boldsymbol{z}}_i^{(s)} = P_s \widetilde{\boldsymbol{z}}_i^{(s)} = \boldsymbol{x}_i^{(s)}.$$

Now, the pairwise Euclidean distance between $\boldsymbol{x}_i^{(s)}$ and $\boldsymbol{x}_j^{(s)}$

$$\|\boldsymbol{x}_i^{(s)} - \boldsymbol{x}_j^{(s)}\|^2 = \|P_s\left(\boldsymbol{z}_i^{(s)} - \boldsymbol{z}_j^{(s)}\right)\|^2 = \|\boldsymbol{z}_i^{(s)} - \boldsymbol{z}_j^{(s)}\|^2 \tag{17}$$

is preserved in the projected space.

In addition, we observed that existing supervised multi-view feature extraction methods such as GMA, MLDA and MvMDA encounter issues, including (i) the embeddings on the training data are often contaminated by noise or outliers; (ii) the generalization to unseen data often leads to different clustering patterns from the training data. These issues will be illustrated later in Fig. 4 on multi-view data mfeat. By introducing orthogonality constraints, all these issues can be addressed as our proposed models will demonstrate. Empirical results via data visualization can be found in Section 6.1.6 in detail.

For multi-view learning, orthogonal projection has already been explored in CCA with two views [30–32] and MCCA with more than two views [33,34] for unsupervised multi-view feature extraction. However, imposing orthogonality constraints has not yet been well studied for supervised multi-view subspace learning. Most existing multi-view subspace learning methods encounter the following two issues when attempting to add orthogonality constraints:

(1) **model incompatibility**. Adding orthogonality constraints may cause incompatibility to inherent constraints already there in existing models. The incompatibility issue can be verified by the following facts. Most of methods [24–26] have their original constraints like (10b), (11), or (12). These

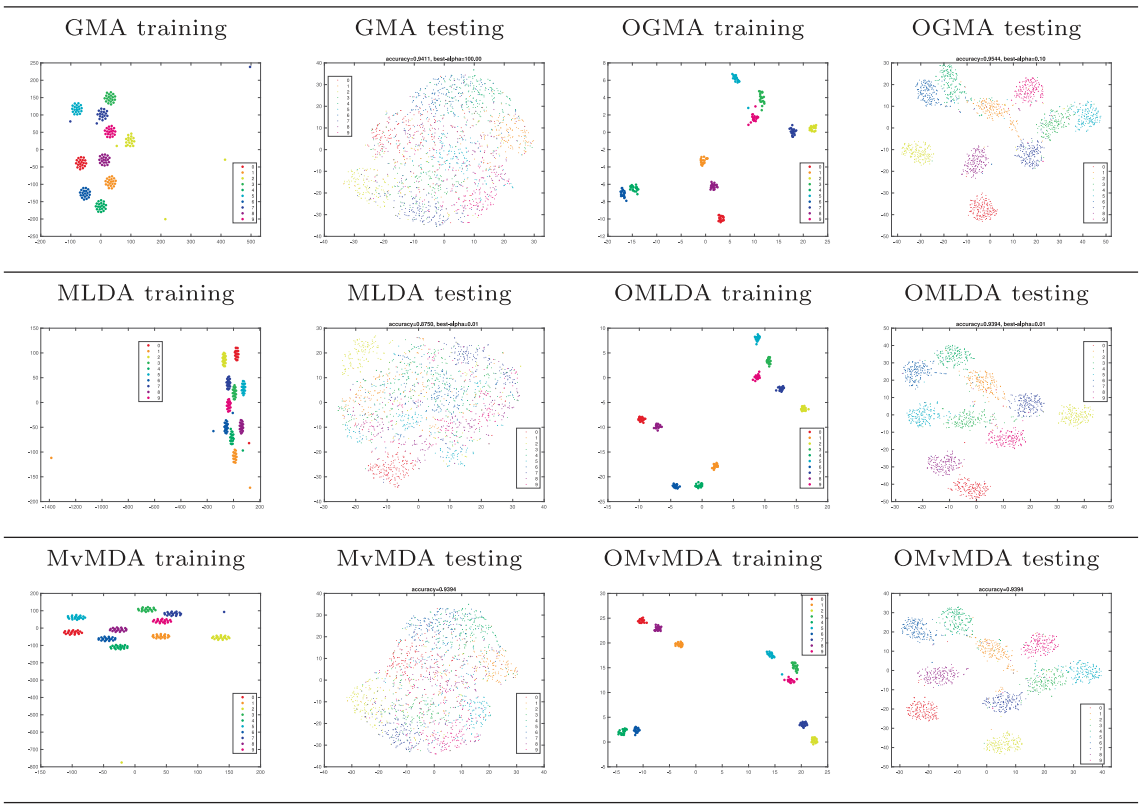


Fig. 4. Data visualization by the 6 methods in the 2-D space via t-SNE on one random split of mfeat (10% training and 90% testing) where the best α of GMA and MvMDA tuned within $\{0.01, 0.1, 1, 10, 100\}$ in terms of accuracy by one-nearest neighbor classifier on the testing sets are used..

constraints may conflict with orthgonality constraints $P_s^{\mathrm{T}}P_s = I_k, \forall s$. To see that, we note that $P_s^{\mathrm{T}}S_{\mathrm{w}}^{(s)}P_s \succeq \lambda_{\min}P_s^{\mathrm{T}}P_s$ where $\lambda_{\min}$ is the smallest eigenvalue of $S_{\mathrm{w}}^{(s)}$, and so if $\lambda_{\min} > 1$, then there is no way to satisfy both constraints at the same time. The same conclusion can be achieved for constraints (11) and (12) after concatenating all projection matrices as $P = [P_1, \ldots, P_s]$.

(2) **optimization difficulty**. Even if there is no incompatibility issue, the resulting optimization problem is generally hard to solve under orthogonality constraints. Generic optimization methods are often too slow even for datasets of modest scale and practically infeasible for high dimensional data. As a result, most existing learning methods [24,25,65] resort to solving certain related relaxed problems of their original formulations as generalized eigenvalue problems, for which well-developed numerical linear algebra techniques can be readily deployed to handle high-dimensional datasets but at a price of degrading learning performance.

4.2. A Trace Ratio Formulation

To address the model incompatibility challenge, we seek to the trace ratio formulation, which has been previously studied in the case of the trace ratio formulation *vs.* the ratio trace formulation for single-view dimensionality reduction in the context of LDA. Authors in [58] argued that the trace ratio formulation with the orthogonality constraint is essential and can lead to superiority over the ratio trace formulation which is a relaxation of the trace ratio formulation as a generalized eigenvalue problem. The cross-correlation between two views in CCA is inherently defined as a trace ratio formulation [20]. Moreover, the objective function of the trace ratio formulation is invariant under any orthogonal transformation, which is more beneficial to classification and clustering in the reduced space than the ratio trace formulation that is invariant under any non-singular transformation. This motivates the study of orthogonal LDA (OLDA) [58,59] and orthogonal CCA (OCCA) [32]. However, no orthogonal extension to supervised multi-view subspace learning has yet been explored.

Motivated by the above observations, we propose a novel trace ratio formulation for unified orthogonal multi-view subspace learning (OMvSL) given by

$$\max_{\{P_s\}} \sum_{s=1}^{v}\sum_{t=1}^{v} \frac{\mathrm{tr}\left(P_s^{\mathrm{T}}\Phi_{s,t}P_t\right)}{\sqrt{\mathrm{tr}\left(P_s^{\mathrm{T}}\Psi_{s,s}P_s\right)P_t^{\mathrm{T}}}\sqrt{\mathrm{tr}\left(P_t^{\mathrm{T}}\Psi_{t,t}P_t\right)}} \tag{18a}$$

$$\text{s.t. } P_s^{\mathrm{T}}P_s = I_k, \forall s, \tag{18b}$$

where $\Psi_{s,s}$ for $s = 1, \ldots, v$ are positive semi-definite matrices. As stated in [58], the trace ratio formation is an essential formulation for general dimensionality reduction and may lead to solutions that are superior to the ones from the ratio trace formulation.

The proposed OMvSL (18) encompasses OLDA and OMCCA as special cases:

1. For $v = 1$, (18) with $\Psi_{1,1} = S_{\mathrm{b}}^{(1)}$ and $\Psi_{1,1} = S_{\mathrm{w}}^{(1)}$ reduces to OLDA.
2. For $v \geqslant 2$, (18) with $\Phi_{s,t} = C_{s,t}$ and $\Psi_{s,s} = C_{s,s}$ becomes OMCCA (8).

OMvSL (18) can be used to inspire various models in the form of trace ratio formulations. We shall present various novel models instantiated from OMvSL (18) for multi-view discriminant analysis

in SubSection 4.3 and multi-view multi-label classification in SubSection 4.4.

OMvSL is a versatile framework, but it presents a difficult optimization problem to solve. Generic optimization techniques [66–68] can always be applied, but they ignore the special form in the objective, are usually not so efficient as customized algorithms, and, worst of all, are not practically feasible even for datasets of modest scale. In Section 5, we will present a successive approximation algorithm that approximately solves OMvSL efficiently to address the optimization difficulty issue.

4.3. Novel Multi-view Discriminant Analysis Models

Three orthogonal multi-view discriminant analysis models are proposed, inspired by existing models similar to (18) for multi-class classification where $\boldsymbol{y}_i \in \{0,1\}^c$ and $\boldsymbol{y}_i^{\mathrm{T}}\mathbf{1}_c = 1$ [24–26]. Each new model is intrinsically different from its corresponding existing model due to the trace ratio formulation (18a) and orthogonality constraints (18b).

Orthogonal GMA. The proposed orthogonal variant of GMA (10), called *Orthogonal GMA* (OGMA), is (18) with

$$\Phi_{s,t} = \begin{cases} S_{\mathrm{b}}^{(s)}, & s = t, \\ \alpha_{s,t} C_{s,t}, & s \neq t, \end{cases} \tag{19a}$$

$$\Psi_{s,s} = S_{\mathrm{w}}^{(s)}. \tag{19b}$$

Orthogonal MLDA. The proposed orthogonal variant of MLDA (10a) with (12), called *Orthogonal MLDA* (OMLDA), is (18) with (19a) and

$$\Psi_{s,s} = C_{s,s}. \tag{20}$$

Orthogonal MvMDA. The proposed orthogonal variant of MvMDA (13), called *Orthogonal MvMDA* (OMvMDA), is (18) with

$$\Phi_{s,t} = A, \Psi_{s,s} = S_{\mathrm{w}}^{(s)}. \tag{21}$$

4.4. Novel Multi-view Multi-label Classification Models

In multi-view multi-label classification, the output $\boldsymbol{y}_i \in \{0,1\}^c$ with c labels and $\left\{\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(v)}, \boldsymbol{y}_i\right\}_{i=1}^n$ is the paired data. Under the proposed formulation (18), we can come up the following two strategies to incorporate output data for multi-view multi-label classification:

Orthogonal Multi-view Multi-label CCA (OM^{2}CCA). This approach is proposed to take the output labels in $Y = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n] \in \{0,1\}^{c \times n}$ as the $(v+1)$st view $X_{v+1} := Y$ in OMCCA [29]. Together with v input views, there are $v+1$ views. OMCCA is employed to learn projection matrices $\{P_s\}$ and $P_{v+1} := P_Y$ in a latent common space, where P_Y is the projection matrix of Y. This idea has been explored for $v = 1$ in [29,30,33]. OMCCA is instantiated from (18) with

$$\Phi_{s,t} = \begin{cases} 0, & s = t, \\ C_{s,t}, & s \neq t, \end{cases} \tag{22a}$$

$$\Psi_{s,s} = C_{s,s}, \tag{22b}$$

for $s, t = 1, \ldots, v+1$, where $C_{s,v+1} = X_s H Y = C_{v+1,s}^{\mathrm{T}}$.

Orthogonal Hilbert–Schmidt Independence Criterion (OHSIC). This approach is proposed to take the HSIC criterion [69] for learning embedding of each input view. The estimator of HSIC is defined as

$$\mathrm{HSIC}(Z_s, Y) = \frac{1}{(n-1)^2} \mathrm{tr}\left(Z_s^{\mathrm{T}} Z_s H_n Y^{\mathrm{T}} Y H_n\right), \tag{23}$$

where $Z_s = P_s^{\mathrm{T}} X_s$ and $Z_s^{\mathrm{T}} Z_s$ is the linear kernel of the projected data of view s. To achieve the best alignment between Z_s and Y, the maximization of HSIC with respect to P_s is expected. The proposed HSIC method is instantiated from (18) with

$$\Phi_{s,t} = \begin{cases} X_s H_n Y^{\mathrm{T}} Y H_n X_s^{\mathrm{T}}, & s = t, \\ \alpha_{s,t} C_{s,t}, & s \neq t, \end{cases} \tag{24a}$$

$$\Psi_{s,s} = C_{s,s}, \tag{24b}$$

for $s, t = 1, \ldots, v$. Different from (22), this approach does not learn P_Y.

5. The Proposed Optimization Algorithm for OMvSL

For ease of presentation, we rewrite OMvSL (18) as

$$\max_{\{P_s\}} g(\{P_s\}) : \text{s.t.} P_s^{\mathrm{T}} P_s = I_k,\ \mathscr{R}(P_s) \subseteq \mathscr{R}(\Psi_{s,s})\ \forall s, \tag{25}$$

where $\mathscr{R}(\Psi_{s,s})$ is the column subspace of $\Psi_{s,s}$ and

$$g(\{P_s\}) := \sum_{s=1}^{v} \sum_{t=1}^{v} \frac{\mathrm{tr}\left(P_s^{\mathrm{T}} \Phi_{s,t} P_t\right)}{\sqrt{\mathrm{tr}\left(P_s^{\mathrm{T}} \Psi_{s,s} P_s\right)} \sqrt{\mathrm{tr}\left(P_t^{\mathrm{T}} \Psi_{t,t} P_t\right)}}.$$

For $k = 1$, all P_s are column vectors. By convention that we use lowercase letters for vectors, we will replace them by $\boldsymbol{p}_s$ instead. Since $g(\{\boldsymbol{p}_s\})$ is homogeneous in each $\boldsymbol{p}_s$, i.e., $g(\{\boldsymbol{p}_s/\alpha_s\}) \equiv g(\{\boldsymbol{p}_s\})$ for any scalar $\alpha_s > 0$, the constraint $\boldsymbol{p}_s^{\mathrm{T}} \boldsymbol{p}_s = 1$ is inconsequential. In fact, (25) is equivalent to

$$\max_{\{\boldsymbol{p}_s \in \mathbb{R}^{d_s}\}} f(\{\boldsymbol{p}_s\}) : \text{s.t.}\ \boldsymbol{p}_s^{\mathrm{T}} \Psi_{s,s} \boldsymbol{p}_s = 1, \boldsymbol{p}_s \in \mathscr{R}(\Psi_{s,s})\ \forall s, \tag{26}$$

where $f(\{\boldsymbol{p}_s\})$ is given by

$$f(\{\boldsymbol{p}_s\}) := \sum_{s=1}^{v} \sum_{t=1}^{v} \boldsymbol{p}_s^{\mathrm{T}} \Phi_{s,t} \boldsymbol{p}_t. \tag{27}$$

The KKT condition of (26) gives rise to a multi-parameter eigenvalue problem:

$$\mathscr{A}\boldsymbol{p} = \mathscr{B}\Lambda\boldsymbol{p},\ \boldsymbol{p} \in \mathscr{R}(\mathscr{B}), \tag{28a}$$

where

$$\mathscr{A} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1v} \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{v1} & \Phi_{v2} & \cdots & \Phi_{vv} \end{bmatrix}, \mathscr{B} = \begin{bmatrix} \Psi_{11} & & & \\ & \Psi_{22} & & \\ & & \ddots & \\ & & & \Psi_{vv} \end{bmatrix}, \tag{28b}$$

$$\Lambda = \begin{bmatrix} \lambda_1 I_{d_1} & & & \\ & \lambda_2 I_{d_2} & & \\ & & \ddots & \\ & & & \lambda_v I_{d_v} \end{bmatrix},\ \boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_v \end{bmatrix}. \tag{28c}$$

This is also a long standing problem in statistics, and there is no existing numerical technique that is readily available to solve it with guarantee. Existing methods include variations of the power method for matrix eigenvalues [70], which are simple to use but often slowly convergent, and adaptations of common optimization techniques onto Riemannian manifolds to solve (26) [71,72], which often converge faster but use the gradient or even Hessian of f and, as a result, are not particularly well suited for large scale problems. None of those methods guarantee to deliver the global optimum of (26).

In many real-world applications, an approximate solution is just as good as a very accurate solution. A relaxed problem to (26) is

$$\max_{\{\boldsymbol{q}_s\}} f(\{\boldsymbol{q}_s\}) : \text{ s.t. } \sum_{s=1}^{v} \boldsymbol{q}_s^{\mathrm{T}} \Psi_{s,s} \boldsymbol{q}_s = 1,\ \boldsymbol{q}_s \in \mathscr{R}(\Psi_{s,s})\ \forall s. \tag{29}$$

The KKT condition for (29) is

$$\mathscr{A}\boldsymbol{q} = \lambda \mathscr{B}\boldsymbol{q},\ \boldsymbol{q} \in \mathscr{R}(\mathscr{B}) \tag{30}$$

which is a generalized eigenvalue problem that has been well studied, where $\mathscr{A}$ and $\mathscr{B}$ are as given by (28b). Often

$$\mathscr{R}(\mathscr{A}) \subseteq \mathscr{R}(\mathscr{B})$$

which we will assume in this paper and the top eigenvector $\boldsymbol{q}$ is the maximizer of (29). Even though $\mathscr{B}$ is positive semi-definite, it is possible that $\mathscr{B}$ is singular. That can cause serious numerical problems and degrade solution effectiveness to the underlying data science application which we will elaborate on in A, where a Krylov subspace projection method, Algorithm 2, will be proposed to solve (30) for its top eigenpair in such a way that singular $\mathscr{B}$ does not matter.

We propose to construct an approximation solution for (26), and thereby for (25) with $k = 1$, from the solution to (29) for $k = 1$ as follows. Let $(\lambda_1, \boldsymbol{q}^{\mathrm{opt}} = [\boldsymbol{q}_s^{\mathrm{opt}}])$ with $\boldsymbol{q}_s^{\mathrm{opt}} \in \mathbb{R}^{d_s}$ be the top eigenpair of the eigenvalue problem (30). An approximate solution is then constructed by

$$\gamma_s = \|\boldsymbol{q}_s^{\mathrm{opt}}\|_2,\ \boldsymbol{p}_s^{\mathrm{opt}} = \boldsymbol{q}_s^{\mathrm{opt}}/\gamma_s,\ \forall s. \tag{31}$$

This solves (25) with $k = 1$ approximately, or finds an approximation to the first columns of optimal P_s of (25). Suppose that approximations to the first ℓ columns, say $\boldsymbol{p}_s^{(j)} \in \mathbb{R}^{d_s}$ for $1 \leqslant j \leqslant \ell$, of nearly optimal P_s of (25) are obtained and $\ell < k$. Let

$$P_s^{(\ell)} = \left[\boldsymbol{p}_s^{(1)}, \boldsymbol{p}_s^{(2)}, \ldots, \boldsymbol{p}_s^{(\ell)}\right] \in \mathbb{R}^{d_s \times \ell},\ \forall s. \tag{32}$$

It is reasonable to assume

$$\left[P_s^{(\ell)}\right]^{\mathrm{T}} P_s^{(\ell)} = I_\ell,\ \mathscr{R}\left(P_s^{(\ell)}\right) \subseteq \mathscr{R}(\Psi_{s,s}),\ \forall s. \tag{33}$$

We propose to find the next columns of nearly optimal P_s for all s of (25) by solving

$$\max_{\{\boldsymbol{q}_s \in \mathbb{R}^{d_s}\}} f(\{\boldsymbol{q}_s\}) : \text{ s.t. } \sum_{s=1}^{v} \boldsymbol{q}_s^{\mathrm{T}} \Psi_{s,s} \boldsymbol{q}_s = 1,\ \boldsymbol{q}_s \in \mathscr{R}(\Psi_{s,s})\ \forall s, \tag{34a}$$

$$\boldsymbol{q}_s^{\mathrm{T}} P_s^{(\ell)} = 0\ \forall s, \tag{34b}$$

and then normalize each $\boldsymbol{q}_s$ of the optimizer of (34) as in (31) to construct the next $\boldsymbol{p}_s^{(\ell+1)}$.

Theorem 1. Given $P_s^{(\ell)}$ as in (32) satisfying (33), problem (34) is equivalent to

$$\max_{\{\boldsymbol{q}_s \in \mathbb{R}^{d_s}\}} f_\ell(\{\boldsymbol{q}_s\}) : \text{ s.t. } \sum_{s=1}^{v} \boldsymbol{q}_s^{\mathrm{T}} \Psi_{s,s}^{(\ell)} \boldsymbol{q}_s = 1,\ \boldsymbol{q}_s \in \mathscr{R}\left(\Psi_{s,s}^{(\ell)}\right)\ \forall s, \tag{35}$$

where

$$\Pi_s^{(\ell)} = I_{d_s} - P_s^{(\ell)}\left[P_s^{(\ell)}\right]^{\mathrm{T}}, \tag{36a}$$

$$\Phi_{s,t}^{(\ell)} = \Pi_s^{(\ell)} \Phi_{s,t} \Pi_t^{(\ell)},\ \Psi_{s,s}^{(\ell)} = \Pi_s^{(\ell)} \Psi_{s,s} \Pi_s^{(\ell)}, \tag{36b}$$

$$f_\ell(\{\boldsymbol{q}_s\}) = \sum_{s,t} \boldsymbol{q}_s^{\mathrm{T}} \Phi_{s,t}^{(\ell)} \boldsymbol{q}_t. \tag{36c}$$

We defer the proof of this theorem to B. In view of our previous discussion, problem (35) is equivalent to finding the top eigenpair of

$$\mathscr{A}^{(\ell)}\boldsymbol{q} = \lambda \mathscr{B}^{(\ell)}\boldsymbol{q} \quad \text{with} \boldsymbol{q} \in \mathscr{R}\left(\mathscr{B}^{(\ell)}\right), \tag{37}$$

where $\mathscr{A}^{(\ell)}$ and $\mathscr{B}^{(\ell)}$ take the same form as $\mathscr{A}$ and $\mathscr{B}$ in (28b), except with all $\Phi_{s,t}$ and $\Psi_{s,s}$ replaced by $\Phi_{s,t}^{(\ell)}$ and $\Psi_{s,s}^{(\ell)}$, respectively. Note now that $\mathscr{B}^{(\ell)}$ is guaranteed singular for $\ell > 1$ because for each s,

$$\begin{aligned} \mathrm{rank}\left(\Psi_{s,s}^{(\ell)}\right) &= \mathrm{rank}\left(\Pi_s^{(\ell)} \Psi_{s,s}^{1/2}\right) \\ &\leqslant \min\left\{\mathrm{rank}\left(\Pi_s^{(\ell)}\right), \mathrm{rank}\left(\Psi_{s,s}^{1/2}\right)\right\} \\ &\leqslant \mathrm{rank}\left(\Pi_s^{(\ell)}\right) = d_s - \ell. \end{aligned}$$

Hence the range constraint $\boldsymbol{q} \in \mathscr{R}\left(\mathscr{B}^{(\ell)}\right)$ is indispensable. Any straightforward application of existing eigen-computation routine to $\mathscr{A}^{(\ell)} - \lambda\mathscr{B}^{(\ell)}$ will likely encounter some numerical issue. But we will again resort to Algorithm 2 in A to solve it. Note that $\boldsymbol{q} \in \mathscr{R}\left(\mathscr{B}^{(\ell)}\right)$ is equivalent to $\boldsymbol{q}_s \in \mathscr{R}\left(\Psi_{s,s}^{(\ell)}\right)\ \forall s$ in (35).

Algorithm 1: OSAVE: Orthogonal Successive Approximation via Eigenvectors

Input: $\left\{\Phi_{s,t} \in \mathbb{R}^{d_s \times d_t},\ 1 \leqslant s, t \leqslant v\right\}$, $\left\{\Psi_{s,s} \in \mathbb{R}^{d_s \times d_s},\ 1 \leqslant s \leqslant v\right\}$, integer $1 \leqslant k \leqslant \min\{d_1, \ldots, d_v\}$;

Output: $\left\{P_s \in \mathbb{O}^{d_s \times k}\right\}$, the set of most correlated matrices.

1: compute the top eigenvector $[\boldsymbol{q}_1^{\mathrm{T}}, \boldsymbol{q}_2^{\mathrm{T}} \ldots, \boldsymbol{q}_v^{\mathrm{T}}]^{\mathrm{T}}$ of $\mathscr{A} - \lambda\mathscr{B}$ by Algorithm 2 in A, where $\boldsymbol{q}_s \in \mathbb{R}^{d_s}$;
2: $\boldsymbol{p}_s^{(1)} = \boldsymbol{q}_s/\|\boldsymbol{q}_s\|_2$ for $s = 1, 2, \ldots, v$;
3: **for** $\ell = 1, 2 \ldots, k-1$ **do**
4: compute the top eigenvector $[\boldsymbol{q}_1^{\mathrm{T}}, \boldsymbol{q}_2^{\mathrm{T}} \ldots, \boldsymbol{q}_v^{\mathrm{T}}]^{\mathrm{T}}$ of $\mathscr{A}^{(\ell)} - \lambda\mathscr{B}^{(\ell)}$ by Algorithm 2 in A, where $\boldsymbol{q}_s \in \mathbb{R}^{d_s}$;
5: $\boldsymbol{p}_s^{(\ell+1)} = \boldsymbol{q}_s/\|\boldsymbol{q}_s\|_2$ for $s = 1, 2, \ldots, v$;
6: **end for**
7: $P_s = \left[\boldsymbol{p}_s^{(1)}, \ldots, \boldsymbol{p}_s^{(k)}\right]$ for $s = 1, 2, \ldots, v$;
8: **return** $\left\{P_s \in \mathbb{O}^{d_s \times k}\right\}$.

Algorithm 1 summarizes our range constrained successive approximation method for solving OMvSL.

5.1. Implementation Details

According to Algorithm 2 in A, the efficiency of Algorithm 1 critically depends on the execution of matrix–vector products by $\mathscr{A}^{(\ell)}$ and $\mathscr{B}^{(\ell)}$. Noting that how $\mathscr{A}^{(\ell)}$ and $\mathscr{B}^{(\ell)}$ are defined, together with (36a) and (36b), we find that

$$\mathscr{A}^{(\ell)} = \Pi^{(\ell)}\mathscr{A}\Pi^{(\ell)},\ \mathscr{B}^{(\ell)} = \Pi^{(\ell)}\mathscr{B}\Pi^{(\ell)},$$

where $\Pi^{(\ell)} = \mathrm{diag}(\Pi_1^\ell, \ldots, \Pi_s^\ell)$. Thus $\boldsymbol{y} := \mathscr{X}^{(\ell)}\boldsymbol{x}$ where $\mathscr{X}$ is either $\mathscr{A}$ or $\mathscr{B}$ can be done in three steps:

$$\boldsymbol{x} \leftarrow \Pi^{(\ell)}\boldsymbol{x}, \tag{38a}$$

$$\boldsymbol{y} \leftarrow \mathscr{X}\boldsymbol{x}, \tag{38b}$$

$$\boldsymbol{y} \leftarrow \Pi^{(\ell)}\boldsymbol{y}. \tag{38c}$$

The operations in (38a) and (38c) are the same one, and should be implemented as follows. In the case of (38a), write $\boldsymbol{x} = [\boldsymbol{x}_1^{\mathrm{T}}, \ldots, \boldsymbol{x}_v^{\mathrm{T}}]^{\mathrm{T}}$ where $\boldsymbol{x}_s \in \mathbb{R}^{d_s}$ and do

$$\boldsymbol{x}_s \leftarrow \boldsymbol{x}_s - P_s^{(\ell)}\left(\left[P_s^{(\ell)}\right]^{\mathrm{T}}\boldsymbol{x}_s\right)\ \forall s,$$

where the bracket must be respected for maximum computational efficiency. The operation in (38b) can be broken into many minions $\Phi_{s,t}\boldsymbol{x}_t, \Psi_{s,s}\boldsymbol{x}_s$ for all s, t whose calculations depend on the struc-

tures in $\Phi_{s,t}$ and $\Psi_{s,s}$ from the underlying task. While it is impossible for us to offer recommendations on a very general setting, a frequent scenario where OMvSL is needed has $\Phi_{s,t}$ and $\Psi_{s,s}$ taking the form

$$\Phi_{s,t} = A_s A_t^{\mathrm{T}}, \quad \Psi_{s,s} = B_s B_s^{\mathrm{T}} \tag{39a}$$

where

$$A_s = A_s^{\mathrm{raw}}\left(I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}\right) \in \mathbb{R}^{d_s\times n}, \tag{39b}$$

$$B_s = B_s^{\mathrm{raw}}\left(I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}\right) \in \mathbb{R}^{d_s\times n}. \tag{39c}$$

Here A_s^{raw} and B_s^{raw} represent raw input data matrices from an application, which may also be sparse. In such a scenario, A_s and B_s should not be formed explicitly in a large scale application, i.e., at least one of d_s and n is large, say in the tens of thousands or more, and neither should $\Phi_{s,t}$ and $\Psi_{s,s}$. As an example, $\boldsymbol{y}_s := \Phi_{s,t}\boldsymbol{x}_t$ can be executed in the order as follows:

$$\boldsymbol{z} \leftarrow \left(A_t^{\mathrm{raw}}\right)^{\mathrm{T}}\boldsymbol{x}_t,\ \boldsymbol{z} \leftarrow \boldsymbol{z} - \frac{\mathbf{1}_n^{\mathrm{T}}\boldsymbol{z}}{n},\ \boldsymbol{y}_s \leftarrow A_s^{\mathrm{raw}}\boldsymbol{z}.$$

5.2. Complexity Analysis

To get a sense of the computational complexity of OSAVE (Algorithm 1), in what follows we present a rough estimate, assuming $\Phi_{s,t}$ and $\Psi_{s,s}$ are given and dense. For the ℓth loop: lines 3–6 of Algorithm 1 which calls Algorithm 2 in A, we have, for the leading cost terms for one loop of Algorithm 2 (lines 6–10),

(a) matrix–vector products by $\mathscr{A}^{(\ell)}$ and $\mathscr{B}^{(\ell)}$: $2n_{n_{\mathrm{kry}}}\left(d^2 + \sum_s d_s^2 + 8d\ell\right)$,
(b) orthgonalization in generating W : $6dn_{n_{\mathrm{kry}}}$ if by the Lanczos process or $2dn_{n_{\mathrm{kry}}}^2$ if also with full reorthgonalization (recommended),
(c) forming $W^{\mathrm{T}}AW$ and $W^{\mathrm{T}}BW$ (assuming AW and BW built along the way are reused): $4dn_{n_{\mathrm{kry}}}^2$,
(d) solving $W^{\mathrm{T}}AW - \lambda W^{\mathrm{T}}BW$: $14n_{n_{\mathrm{kry}}}^3$ [p.500] [38].

Here $d = \sum_s d_s$ and these estimates work for $\ell = 0$, i.e., line 1 of Algorithm 1, too. For simplicity, let us assume that on average Algorithm 2 takes m iterations to finish, and full reorthgonalization is used for robustness. Then the overall complexity estimate is

$$m\left\{kn_{n_{\mathrm{kry}}}\left[2d^2 + \sum_s d_s^2 + 6dn_{n_{\mathrm{kry}}}\right] + 8n_{n_{\mathrm{kry}}}dk^2\right\} \approx 2mkn_{n_{\mathrm{kry}}}d^2, \tag{40}$$

where we have dropped the cost in solving $W^{\mathrm{T}}AW - \lambda W^{\mathrm{T}}BW$ due to that $n_{n_{\mathrm{kry}}}$ is usually of $O(1)$, and we have assumed $k \ll d$ in practice. Further improvement in complexity is possible if A_s and B_s in (39) are very sparse, and then d^2 in (40) can be replaced by the total number of nonzero entries in A_s and B_s for all s. For the ease of comparison, we summarize the computational complexity of several related methods in Table 1, where $O\left(d^2n\right)$ in most of the complexity estimates is for forming all $\Phi_{s,t}$ and $\Psi_{s,s}$. Note that n_{kry} is small, e.g., 10 as used in our implementation and the number m of iterations in OSAVE is usually small, e.g., around 10. We include a parameter m (the number of iterations) in the complexity for SaDCCA as it has to be solved iteratively and it is often rather large for a reasonable precision requirement. It is clear that OSAVE has complexity depending only on d^2 instead of d^3 of others except RAMC. Hence OSAVE is more efficient for high-dimensional data and more views.

Table 1
Computational complexity where $d = \sum_{s=1}^{v} d_s$, m is the number of iterations for those that are solved iteratively, k is the reduced dimension, c is the number of class labels, and n_{kry} is the order of the Krylov space in OSAVE.

method	complexity
GMA	$O\left(d^3 + d^2n\right)$
MvMDA	$O\left(d^3 + d^2n\right)$
MLDA	$O\left(d^3 + d^2n\right)$
MULDA	$O\left(md^3 + d^2n\right)$
SaDCCA	$O\left(md^2n\right)$
CRMvFE	$O\left(d^3 + d^2n\right)$
RAMC	$O\left(m\left[nc + \min\left\{d^3 + nd^2, n^3 + n^2d\right\}\right]\right)$
OSAVE	$O\left(mkn_{\mathrm{kry}}d^2 + d^2n\right)$

6. Experiments

In this section, we will evaluate the effectiveness of our proposed models instantiated from the unified framework (18) by comparing with existing methods on two learning tasks: multi-view feature extraction in SubSection 6.1 and multi-view multi-label classification in SubSection 6.2.

6.1. Multi-view Feature Extraction

6.1.1. Datasets

Five datasets in Table 2 are used to evaluate the performance of the proposed models: OGMA, OMLDA, and OMvMDA in terms of multi-view feature extraction. We apply various feature descriptors, including CENTRIST [73], GIST [74], LBP [75], histogram of oriented gradient (HOG), color histogram (CH), and SIFT-SPM [76], to extract features of views for image datasets: Caltech101[1][77] and Scene15[2] [76]. Note that we drop CH for Scene15 due to the gray-level images. Multiple Features (mfeat)[3] and Internet Advertisements (Ads)[4] are publicly available from UCI machine learning repository. The dataset mfeat contains handwritten numeral data with six views including profile correlations (fac), Fourier coefficients of the character shapes (fou), Karhunen-Love coefficients (kar), morphological features (mor), pixel averages in 2×3 windows (pix), and Zernike moments (zer). Ads is used to predict whether or not a given hyperlink (associated with an image) is an advertisement and has three views: features based on the terms in the images URL, caption, and alt text (url + alt + caption), features based on the terms in the URL of the current site (origurl), and features based on the terms in the anchor URL (ancurl).

6.1.2. Compared Methods

As shown in SubSection 4.3, our proposed models, although instantiated from the proposed framework (18), are inspired by some of the existing ones. Hence, the three proposed models have close counterparts via solving generalized eigenvalue problems. Specifically, the compared methods are.

- GMA [24]: (10a) with constraint (11);
- MLDA and MLDA-m with modifications [26]: (10a) with constraint (12) and its variant;
- MvMDA [25]: (13);
- MULDA and MULDA-m with modifications [26]: MLDA and

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[2] https://figshare.com/articles/15-Scene_Image_Dataset/7007177
[3] https://archive.ics.uci.edu/ml/datasets/Multiple + Features
[4] https://archive.ics.uci.edu/ml/datasets/internet + advertisements

Table 2
Datasets for feature extraction (followed by classification), where the number of features for each view is shown inside the bracket.

Dataset	n	c	view 1	view 2	view 3	view 4	view 5	view 6
mfeat	2000	10	fac (216)	fou (76)	kar (64)	mor (6)	pix (240)	zer (47)
Caltech101-7	1474	7	CENTRIST (254)	GIST (512)	LBP (1180)	HOG (1008)	CH (64)	SIFT-SPM (1000)
Caltech101-20	2386	20	CENTRIST (254)	GIST (512)	LBP (1180)	HOG (1008)	CH (64)	SIFT-SPM (1000)
Scene15	4310	15	CENTRIST (254)	GIST (512)	LBP (531)	HOG (360)	SIFT-SPM (1000)	-
Ads	3279	2	url + alt + caption (588)	origurl (495)	ancurl (472)	-	-	-

MLDA-m with additional uncorrelated constraints, respectively;
- CRMvFE and RCRMvFE [10]: (9) and its variant by replacing square loss with the $\ell_{2,1}$ norm;
- SaDCCA [9]: (14);
- RAMC [42]: (15);
- OGMA: proposed model instantiated from (18) with (19);
- OMLDA: proposed model instantiated from (18) with (19a) and (20);
- OMvMDA: proposed model instantiated from (18) with (21).

Except for MvMDA and OMvMDA, all methods share the same trade-off parameter to balance the pairwise correlation and supervised information. In our experiments, we set $\alpha_{s,t} = \alpha, \forall s \neg = t$ so as to reduce the complexity of model selection and tune $\alpha \in \{0.01, 0.1, 1, 10, 100\}$ for proper balance in supervised setting. To prevent the singularity of matrices $\{\Psi_{s,s}\}$, we add a diagonal matrix with a small value, e.g., 10^{-8}, to $\Psi_{s,s}\ \forall s$ for all compared methods.

6.1.3. Classification

To evaluate the learning performance of compared methods, the 1-nearest neighbor classifier as the base classifier is employed. We run each method to learn projection matrices by varying the dimension of the common subspace $k \in [2, 30]$ for all datasets except for mfeat with $k \in [2, 6]$ due to the smallest view of 6 features. We split the data into training and testing with ratio 10/90. The learned projection matrices are used to transform both training and testing data into the latent common space, and then classifier is trained and tested in this space. Following [34,33,30], the serial feature fusion strategy is employed by concatenating projected features from all views. Classification accuracy is used to measure the learning performance. Experimental results are reported in terms of the average and standard deviation over 10 randomly drawn splits.

Table 3 shows the best results of 13 compared methods on 5 multi-view datasets with 10% training and 90% testing over all tested ks and αs (the analysis on parameter sensitivity and training sample size will be discussed in SubSections 6.1.4 and 6.1.5, respectively). From Table 3, we have the following observations: (i) our proposed models instantiated from (18) generally outperform their counterparts which resort to relax their respective original problems to generalized eigenvalue problems for the convenience of their numerical computations; (ii) our proposed models instantiated from (18) outperform the four most recently methods; (iii) three proposed models produce best results on different datasets, while OGMA and OMLDA perform consistently better than OMvMDA on four of the five datasets. This empirically shows that the model hypothesis in each model is data-dependent, but our proposed trace ratio formulation with orthogonality constraints can help boost the performance of their counterparts with large margins over three of the five datasets.

6.1.4. Parameter Sensitivity Analysis

The sensitivity analyses on parameters k and α are performed by varying one of them while recording the best average accuracy over the other within its testing range.

Fig. 1 shows the results of 13 methods on 5 datasets as k varies. Most compared methods demonstrate the increasing trend in accuracy when k increases. The proposed methods produce consistently better accuracies than others. On Ads, Caltech101-7 and Reuters, our methods show the saturation on accuracy, while MvMDA shows a significant drop after the certain k on three of five datasets.

We further investigate the impact of parameter α on GMA, OGMA, MLDA and OMLDA except MvMDA and OMvMDA since both methods do not contain parameter α. In Fig. 2, GMA and OGMA demonstrates quite robust to α, and the best accuracy can be obtained around $\alpha = 10^{-2}$. However, MLDA and OMLDA are quite sensitive to α and the accuracy decreases significantly especially for $\alpha > 0.1$. These observations imply that more contribution from pairwise correlation may hurt MLDA and OMLDA, but no noticeable impact on GMA and OGMA. Over all tested αs, our proposed methods outperform their counterparts.

6.1.5. Impact on Training Sample Size

We further show the impact of training sample size on the compared methods by varying the ratio of training data from 10% to 60%. The best average results over 10 randomly drawn

Table 3
Means and standard deviations of accuracy by the 1-nearest neighbor classifier on embeddings by 13 methods on 5 multi-view datasets over 10 random draws (10% training and 90% testing). N/A in RCRMvPE for Ads is due to its numerical difficulty in producing a result.

method	mfeat	Ads	Scene15	Caltech101-7	Caltech101-20
GMA	0.9399 ± 0.0087	0.9261 ± 0.0176	0.6166 ± 0.0120	0.9325 ± 0.0104	0.8130 ± 0.0106
MLDA	0.9284 ± 0.0052	0.9309 ± 0.0079	0.5468 ± 0.0137	0.9229 ± 0.0079	0.7659 ± 0.0117
MvMDA	0.9378 ± 0.0091	0.7796 ± 0.0360	0.6088 ± 0.0146	0.9265 ± 0.0078	0.8050 ± 0.0132
MULDA	0.9523 ± 0.0046	0.9249 ± 0.0352	0.5789 ± 0.0121	0.9265 ± 0.0083	0.8220 ± 0.0109
MLDA-m	0.9309 ± 0.0079	0.9418 ± 0.0061	0.5699 ± 0.0120	0.8978 ± 0.0098	0.7377 ± 0.0114
MULDA-m	0.9512 ± 0.0044	0.9282 ± 0.0362	0.5795 ± 0.0154	0.9259 ± 0.0099	0.8217 ± 0.0058
CRMvFE	0.9545 ± 0.0032	0.9312 ± 0.0058	0.6190 ± 0.0061	0.9350 ± 0.0089	0.8251 ± 0.0095
RCRMvFE	0.9402 ± 0.0089	N/A	0.6378 ± 0.0090	0.9310 ± 0.0081	0.8198 ± 0.0069
SaDCCA	0.8963 ± 0.0105	0.8930 ± 0.0127	0.6307 ± 0.0213	0.8935 ± 0.0127	0.7726 ± 0.0105
RAMC	0.9008 ± 0.0093	0.9257 ± 0.0122	0.6268 ± 0.0598	0.9278 ± 0.0085	0.8241 ± 0.0103
OGMA (proposed)	**0.9609 ± 0.0060**	0.9412 ± 0.0114	0.7359 ± 0.0156	**0.9501 ± 0.0052**	0.8600 ± 0.0103
OMLDA (proposed)	0.9571 ± 0.0064	0.9410 ± 0.0115	**0.7547 ± 0.0105**	0.9498 ± 0.0048	**0.8685 ± 0.0100**
OMvMDA (proposed)	0.9599 ± 0.0063	**0.9423 ± 0.0103**	0.7198 ± 0.0191	0.9471 ± 0.0072	0.8428 ± 0.0102

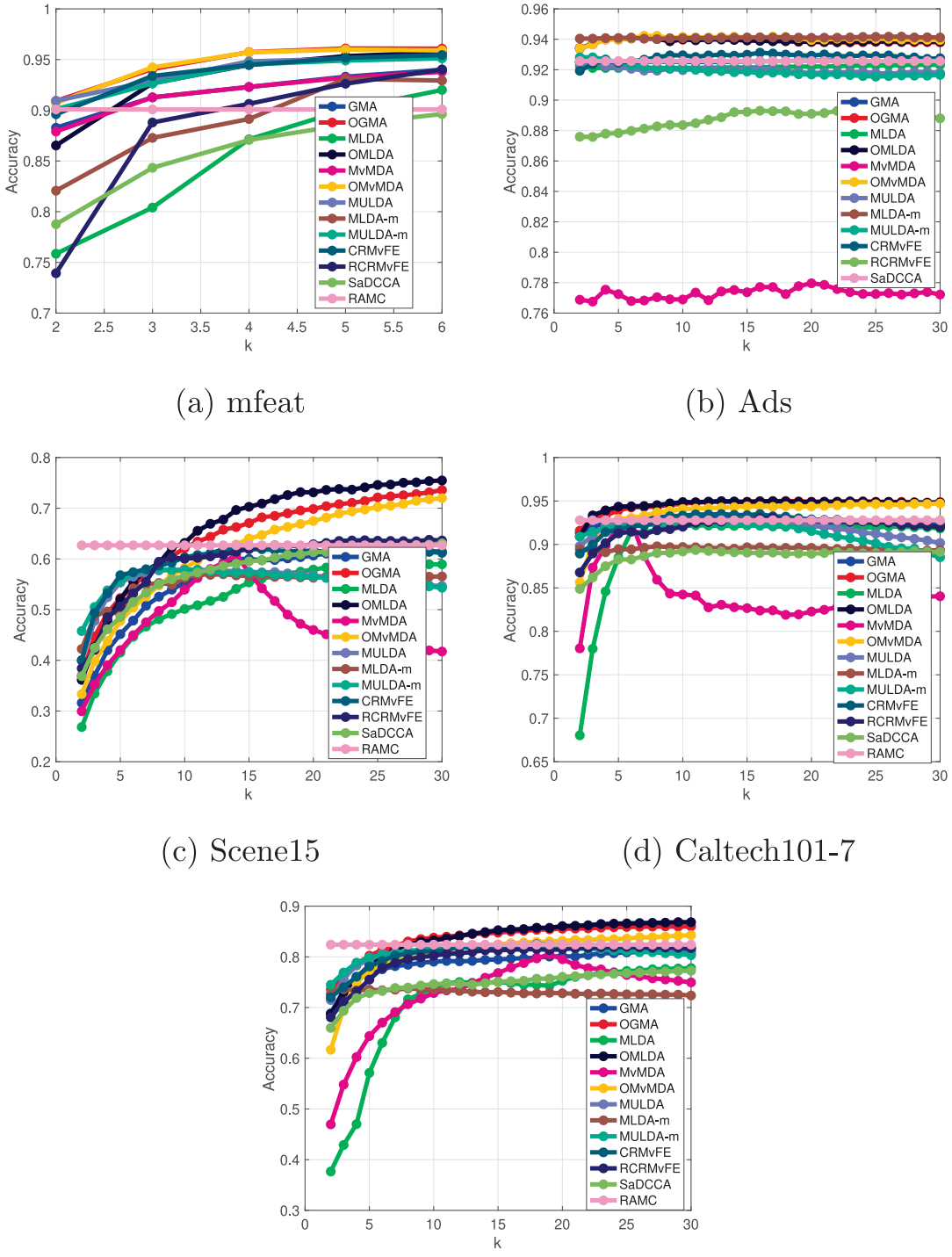


Fig. 1. Classification accuracy of 13 methods on 5 datasets over 10 random splits (10% training and 90% testing), as k varies..

splits are reported. Fig. 3 shows the accuracy improves when the training ratio is increasing on Ads and Caltech101-7. It is observed that (i) all methods show better performance when training sample size increases, (ii) our proposed methods show consistently better results than others, and (iii) all methods converge to similar results when training sample size becomes very large except MvMDA.

6.1.6. Exploratory Analysis via Data Visualization

We further investigate the embeddings learned by our proposed methods and their counterparts, especially for the impact of orthogonality constraints, including three existing methods: GMA, MLDA and MvMDA, and three newly proposed methods: OGMA, OMLDA and OMvMDA. We randomly draw 10% instances from mfeat for training, and the rest 90% for testing. Each method is used to learn projection matrices from training data, and then

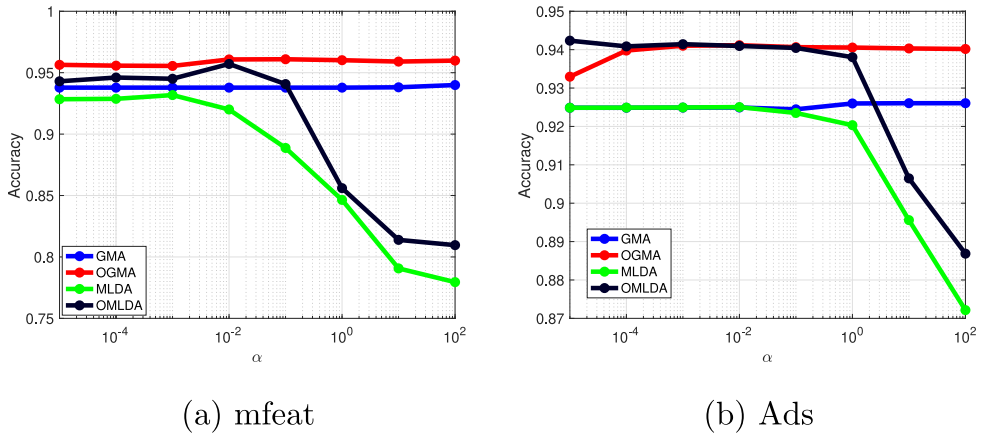


Fig. 2. Classification accuracy by 4 methods on mfeat and Ads over 10 random splits (10% training and 90% testing), as α varies in $\left[10^{-5}, 10^{2}\right]$.

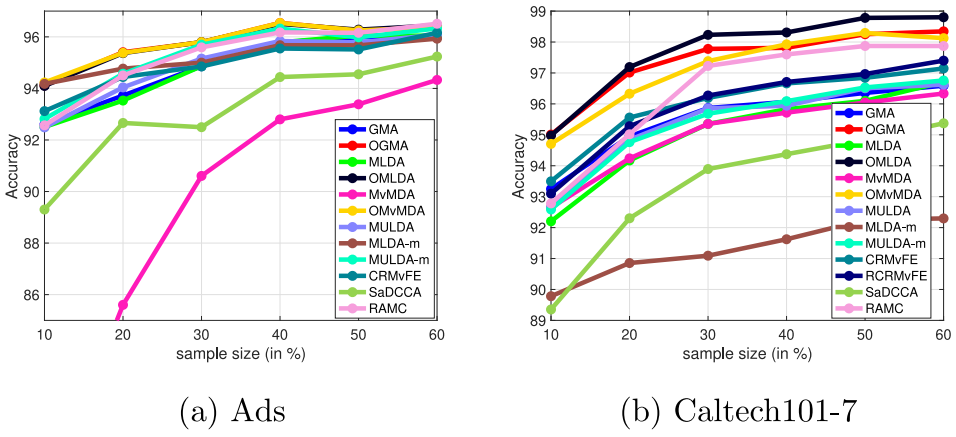


Fig. 3. Classification accuracy by all 13 methods on Ads and Scene15 as the ratio of training data varies from 10% to 60%..

Table 4
Multi-view multi-label datasets for classification

	samples (n)	labels	views (v)
emotions	593	6	2
Corel5k	4999	260	7
espgame	20770	268	7
pascal07	9963	20	7

transforms both training and testing data to the common space $\mathbb{R}^{k}$. The concatenation of projected points for each instance in all 6 views is used as the low-dimensional representation of the instance. t-SNE [78] is used to obtain the 2-D embeddings of the low-dimensional representations for training and testing sets, respectively. Except MvMDA and OMvMDA, the other four methods have a hyperparameter α, which is tuned with the set $\{0.01, 0.1, 1, 10, 100\}$ for the best testing accuracy. The 2-D embeddigns of six methods on both training and testing sets are shown in Fig. 4. We have the following observations: (i) methods without orthogonality constraints suffer from noisy or outliers in embeddings of training data, while methods with orthogonality constraints do not; (ii) the generalization performance in terms of both accuracy and visual pattern of clustering structure by our methods which have orthogonality constraints are superior to their counterparts. These observations are consistent with our motivation we laid out in Section 4.1, that is that orthogonality constraints can admit robustness to data noise and are advantageous for data visualization, and also possess better generalization performance.

Table 5
Results in terms of 7 metrics on emotions over 10 random splits (10% for training and 90% for testing). Best results are in bold.

method	Hamming Loss ↓	One Error ↓	Coverage ↓	Average Precision ↑	Accuracy ↑	macroF1 ↑	microF1 ↑
view-1	0.3060±0.0156	0.4672±0.0312	2.4903±0.1790	0.6647±0.0181	0.2900±0.0612	0.3164±0.0454	0.3971±0.0678
view-2	0.3403±0.0247	0.5949±0.0422	3.1069±0.0625	0.5678±0.0174	0.1832±0.0320	0.2010±0.0454	0.2696±0.0410
concat	0.3046±0.0155	0.4869±0.0359	2.8039±0.1208	0.6290±0.0232	0.2476±0.0462	0.2661±0.0466	0.3557±0.0410
sM2CP	0.3770±0.0298	0.6315±0.0383	3.0390±0.2060	0.5552±0.0275	0.2273±0.0225	0.2840±0.0576	0.3227±0.0407
MCCA	0.3661±0.0267	0.6399±0.0321	3.1830±0.1291	0.5468±0.1291	0.1760±0.0651	0.1953±0.0724	0.2546±0.0771
OM^{2}CCA	0.3006±0.0124	0.4948±0.0488	2.5740±0.1779	0.6492±0.1777	0.2740±0.0512	0.3412±0.0580	0.3949±0.0578
HSIC-GEV	0.3646±0.0241	0.6223±0.0466	3.0798±0.1888	0.5553±0.1888	0.2561±0.0330	0.2470±0.0426	0.3363±0.0350
OHSIC	**0.2953±0.0110**	**0.4655±0.0342**	**2.4850±0.1222**	**0.6662±0.1222**	**0.3116±0.0380**	**0.3554±0.0476**	**0.4325±0.0359**

Table 6
Results in terms of 7 metrics on Corel5k over 10 random splits (10% for training and 90% for testing). Best results are in bold.

method	Hamming Loss ↓	One Error ↓	Coverage ↓	Average Precision ↑	Accuracy ↑	macroF1 ↑	microF1 ↑
view-1	0.0131±0.0001	0.7153±0.0147	95.3444±1.4930	0.2637±0.0055	0.0177±0.0094	0.0074±0.0042	0.0351±0.0185
view-2	0.0131±0.0001	0.7031±0.0110	94.9287±1.6843	0.2689±0.0047	0.0190±0.0063	0.0088±0.0042	0.0374±0.0131
view-3	0.0131±0.0001	0.6606±0.0072	95.3894±1.7478	0.2862±0.0035	0.0322±0.0066	0.0082±0.0017	0.0604±0.0123
view-4	0.0131±0.0000	0.7187±0.0154	97.7932±1.6951	0.2592±0.0045	0.0137±0.0071	0.0048±0.0020	0.0259±0.0133
view-5	0.0131±0.0000	0.7366±0.0107	96.2485±1.4062	0.2502±0.0039	0.0099±0.0035	0.0037±0.0018	0.0195±0.0064
view-6	0.0131±0.0000	0.7365±0.0137	96.2007±1.3439	0.2520±0.0060	0.0103±0.0052	0.0050±0.0023	0.0209±0.0110
view-7	0.0131±0.0000	0.6906±0.0065	96.3108±1.3974	0.2716±0.0035	0.0137±0.0052	0.0059±0.0018	0.0265±0.0099
concat	0.0131±0.0001	0.6591±0.0135	92.5057±2.1126	0.2999±0.0063	0.0291±0.0083	0.0135±0.0039	0.0556±0.0156
sM2CP	0.0131±0.0000	0.7799±0.0109	105.1170±1.4039	0.2120±0.0024	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000
MCCA	0.0131±0.0000	0.7799±0.0115	104.9648±1.4837	0.2121±1.4837	0.0000±0.0000	0.0000±0.0000	0.0000±0.0000
OM^{2}CCA	**0.0130±0.0000**	0.6982±0.0106	94.7535±1.4380	0.2729±1.4651	0.0244±0.0080	0.0126±0.0045	0.0487±0.0158
HSIC-GEV	0.0131±0.0000	0.7885±0.0161	104.6444±1.5763	0.2011±1.6329	0.0169±0.0247	0.0010±0.0017	0.0270±0.0393
OHSIC	**0.0130±0.0001**	**0.6374±0.0126**	**91.8414±1.5051**	**0.3022±1.3774**	**0.0879±0.0092**	**0.0230±0.0032**	**0.1538±0.0136**

Table 7
Results in terms of 7 metrics on espgame over 10 random splits (10% for training and 90% for testing). Best results are in bold.

method	Hamming Loss ↓	One Error ↓	Coverage ↓	Average Precision ↑	Accuracy ↑	macroF1 ↑	microF1 ↑
view-1	0.0174±0.0000	0.6762±0.0052	134.8974±0.5372	0.2235±0.0014	0.0216±0.0032	0.0042±0.0005	0.0340±0.0051
view-2	0.0174±0.0000	0.6766±0.0058	134.6899±0.6207	0.2238±0.0013	0.0214±0.0041	0.0044±0.0006	0.0347±0.0061
view-3	0.0175±0.0000	0.7213±0.0049	129.8373±0.4775	0.2185±0.0015	0.0082±0.0033	0.0048±0.0010	0.0176±0.0067
view-4	0.0175±0.0000	0.7169±0.0032	129.1738±0.7794	0.2201±0.0013	0.0084±0.0017	0.0080±0.0014	0.0183±0.0038
view-5	0.0174±0.0000	0.6668±0.0051	135.3101±0.4779	0.2262±0.0016	0.0229±0.0016	0.0043±0.0004	0.0351±0.0028
view-6	0.0174±0.0000	0.6687±0.0033	135.4435±0.4592	0.2252±0.0010	0.0225±0.0035	0.0043±0.0006	0.0348±0.0057
view-7	0.0175±0.0000	0.7279±0.0049	130.7208±0.5104	0.2160±0.0016	0.0079±0.0022	0.0046±0.0006	0.0171±0.0044
concat	0.0175±0.0000	0.6989±0.0063	**128.8904±0.6606**	0.2283±0.0010	0.0117±0.0021	0.0087±0.0017	0.0254±0.0047
SM2CP	0.0177±0.0002	0.6981±0.0263	136.3555±1.1158	0.2075±0.0079	0.0670±0.0142	0.0221±0.0090	0.1256±0.0233
MCCA	**0.0174±0.0001**	0.6784±0.0518	134.1460±1.9614	0.2249±1.9531	0.0190±0.0112	0.0045±0.0027	0.0306±0.0191
OM^{2}CCA	**0.0174±0.0000**	0.6283±0.0040	132.0874±0.5329	0.2454±0.5074	0.0306±0.0035	0.0068±0.0006	0.0490±0.0055
HSIC-GEV	**0.0174±0.0000**	0.6236±0.0053	131.9247±0.6241	0.2481±0.6241	**0.0900±0.0026**	**0.0244±0.0013**	**0.1606±0.0039**
OHSIC	**0.0174±0.0000**	**0.6207±0.0053**	131.5208±0.5965	**0.2495±0.5965**	0.0362±0.0036	0.0077±0.0007	0.0604±0.0061

Table 8
Results in terms of 7 metrics on pascal07 over 10 random splits (10% for training and 90% for testing). Best results are in bold.

method	Hamming Loss ↓	One Error ↓	Coverage ↓	Average Precision ↑	Accuracy ↑	macroF1 ↑	microF1 ↑
view-1	0.0730±0.0005	0.5946±0.0029	6.9247±0.1447	0.4425±0.0029	0.0686±0.0256	0.0240±0.0074	0.1299±0.0424
view-2	0.0729±0.0002	0.5950±0.0031	6.7332±0.1332	0.4466±0.0033	0.0744±0.0231	0.0246±0.0036	0.1397±0.0364
view-3	0.0715±0.0005	0.5819±0.0044	5.9969±0.1021	0.4800±0.0043	0.0824±0.0238	0.0368±0.0101	0.1500±0.0401
view-4	0.0702±0.0003	0.5656±0.0042	5.8909±0.0956	0.4928±0.0034	0.1320±0.0252	0.0574±0.0071	0.2247±0.0337
view-5	0.0716±0.0004	0.5941±0.0026	6.7623±0.0936	0.4482±0.0032	0.0950±0.0247	0.0284±0.0056	0.1720±0.0387
view-6	0.0719±0.0006	0.5945±0.0022	6.7054±0.0993	0.4498±0.0023	0.0977±0.0270	0.0299±0.0058	0.1764±0.0403
view-7	0.0699±0.0005	0.5617±0.0049	5.6492±0.0718	0.5006±0.0040	0.1206±0.0196	0.0505±0.0064	0.2110±0.0277
concat	0.0700±0.0003	0.5634±0.0061	5.7465±0.1130	0.4996±0.0047	0.1405±0.0165	0.0656±0.0096	0.2385±0.0230
sM2CP	0.1198±0.0165	0.7706±0.0301	9.8902±0.5857	0.3023±0.0419	0.1255±0.0251	**0.0704±0.0119**	0.1974±0.0328
MCCA	0.0691±0.0002	0.5700±0.0054	5.5241±0.0599	0.4991±0.05993	0.0935±0.0571	0.0203±0.0098	0.1553±0.0837
OM^{2}CCA	0.0694±0.0003	0.5723±0.0060	5.5003±0.0788	0.4960±0.0714	0.1268±0.0220	0.0473±0.0078	0.2231±0.0288
HSIC-GEV	**0.0678±0.0004**	**0.5569±0.0046**	5.4652±0.1018	**0.5088±0.1018**	0.1446±0.0134	0.0479±0.0071	0.2357±0.0126
OHSIC	**0.0678±0.0004**	0.5604±0.0046	**5.3753±0.0679**	0.5073±0.0525	**0.1816±0.0116**	0.0681±0.0115	**0.2908±0.0131**

6.2. Multi-view Multi-label Classification

6.2.1. Datasets

The statistics of four publicly available datasets are shown in Table 4, and they are employed to evaluate the proposed methods for multi-view multi-label classification. Dataset emotions[5] has two feature views: 8 rhythmic attributes and 64 timbre attributes. Corel5k [79] is a benchmark dataset for keyword based image retrieval and image annotation. Dataset espgame [80] is obtained from an online game where two players gain points by agreeing on words describing the image. Dataset pascal07 [81] is collected from the Flickr website. The last three datasets have been preprocessed with various feature descriptors and are publicly available[6] [82,83]. In our experiments, we choose 7 descriptors: DenseHue (100), DenseHueV3H1 (300), DenseSift (1000), Gist (512), HarrisHue (100), HarrisHueV3H1 (300), and HarrisSift (1000).

6.2.2. Compared Methods

We compare the following multi-view subspace learning approaches for multi-label classification:

- view-*s*: the embeddings are obtained by PCA on the *s*th view.
- concat: the concatenation of embeddings of all views by PCA.
- MCCA [21]: the output labels considered as an additional view. Hence, there are $v+1$ views. The projection matrix for the output labels is learned but not used.
- sM2PC [51]: (16) with supervised information encoded in the CCA-based model.
- HSIC-GEV: proposed model solved as a generalized eigenvalue problem, which is similar to MLDA, but $\Phi_{s,s}$ is defined in (24a) catering for multi-label outputs.

[5] http://mulan.sourceforge.net
[6] http://lear.inrialpes.fr/people/guillaumin/data.php

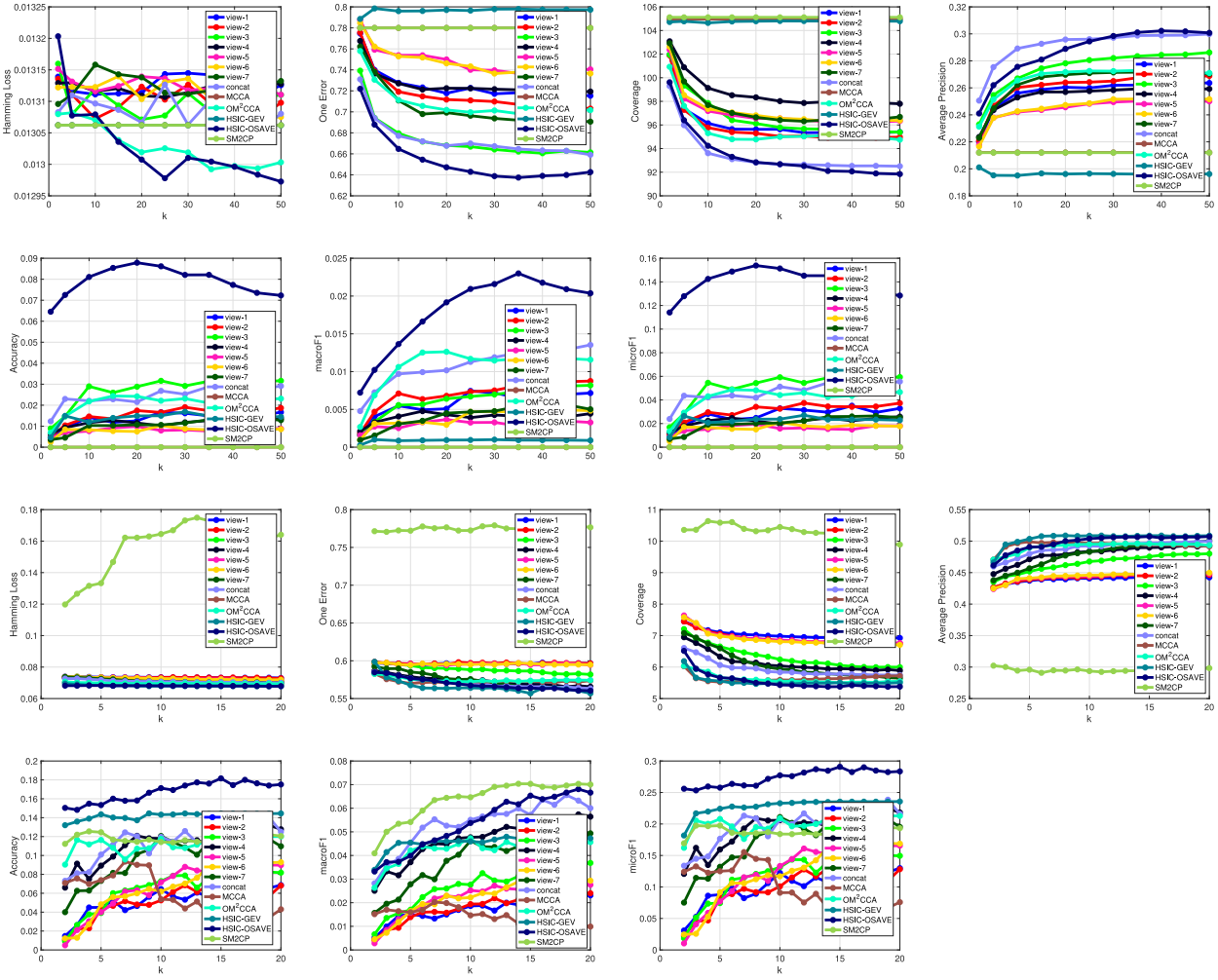


Fig. 5. Results with respect to seven metrics by compared methods on Corel5k (first and second rows) and pascal07 (third and fourth rows) over 10 random splits (10% training and 90% testing), as k varies..

- OM^{2}CCA: the proposed model instantiated from (18) with $v+1$ views using (22). Different from [33], all multiple views as input are used.
- OHSIC: proposed model instantiated from (18) with (4).

After the projection matrices are learned, we apply ML-kNN[7] in the common space as the backend multi-label classifier [36], which has demonstrated good performance over various datasets.

6.2.3. Performance Evaluation

Seven widely-used metrics are used to measure performance, including Hamming Loss, One Error, Coverage, Average Precision, Accuracy, macroF1 and microF1. Each evaluates the performance of a multi-label predictor from different aspects. Their concrete definitions can be found in [44,84]. In particular, the larger the last four metrics are, the better the performance, while for the other three metrics, the smaller the value the better the performance. Following [36], for each method we report the best results and their standard deviations over 10 random training/testing splits in each of the five metrics.

Results by compared methods are shown in Tables 5–8, in which the best results are reported by tuning $\alpha \in \{0.01, 0.1, 1, 10, 100\}$ and $k \in \{2, 5:5:50\}$ except for emotions and pascal07 (MCCA and OM^{2}CCA cannot have k larger than the number of labels), over 10 random splits of 10% training and 90% testing. It can be observed that (i) the joint subspace learning methods generally work better than PCA and the concatenation of individually projected views by PCA; (ii) the proposed OHSIC consistently outperform others except in some cases that sM2CP works best on pascal07 in terms of macroF1 and concat works best on espgame in terms of Coverage.

We further investigate the impact of parameter k on each of the four metrics. Fig. 5 shows the trends of four metrics on Corel5k and pascal07 as k varies. It is observed that a large k generally leads to better performance for all methods, as it should be. Although Hamming Loss on Corel5k shows some fluctuation, the absolute difference is negligibly in the order of 10^{-5}. In summary, OHSIC can work consistently well over all tested ks.

7. Conclusions

In this paper, we start by proposing a trace ratio formulation for multi-view subspace learning, which aims to learn a set of orthogonal projections for desirable advantages such as more noise-tolerant, better suited for data visualization and distance preservation. The proposed formulation can be easily extended for single-view and multi-view learning in the settings of both unsupervised

[7] http://lamda.nju.edu.cn/files/MLkNN.rar

and supervised learning. An efficient successive approximations via eigenvectors method (OSAVE) is designed to approximately solve the optimization problem resulted from the proposed formulation. It is built upon well developed numerical linear algebra technique and can handle large scale datasets. To verify the capability of the proposed formulation and the approximate optimization method, we showcase six new models for two learning tasks. Experimental results on various real-world datasets demonstrate that our proposed models solved by our OSAVE perform competitively to and often better than the baselines.

CRediT authorship contribution statement

Li Wang: Conceptualization, Methodology, Writing - original draft, Supervision, Investigation, Software, Data curation, Validation, Visualization. **Lei-Hong Zhang:** Formal analysis, Writing - original draft. **Chungen Shen:** Writing - review & editing. **Ren-Cang Li:** Formal analysis, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Li Wang is supported in part by NSF DMS-2009689. Lei-Hong Zhang is supported in part by the National Natural Science Foundation of China NSFC-11671246 and NSFC-12071332. Ren-Cang Li is supported in part by NSF DMS-1719620 and DMS-2009689.

Appendix A. An Eigenvalue Algorithm

Currently there is no numerically efficient method to solve OMvSL (18), especially for high-dimensional datasets. Our orthogonal successive approximation via eigenvectors (OSAVE), Algorithm 1 in Section 5, relies upon a Krylov subspace method that is suitable for computing the top eigenpair for the generalized eigenvalue problem. To simplify notation, we will describe the method generically for

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \quad \text{with} \quad \boldsymbol{x} \in \mathscr{R}(B), \tag{A.1}$$

where $A, B \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}}$ are symmetric, the column subspace $\mathscr{R}(A) \subseteq \mathscr{R}(B), B \succeq 0$ and possibly B is singular. Suppose that matrix–vector products, $A\boldsymbol{x}$ and $B\boldsymbol{x}$ for any given $\boldsymbol{x}$, are the only operations that can be done numerically.

The Krylov subspace method will serve as the workhorse of OSAVE that approximately solves OMvSL (18). It is worth noting that B may be singular and will be singular in our applications. A common past practice in data science is simply to perturb B to $B + \epsilon I_d$ for some tiny $\epsilon > 0$ as a regularization and solve $A\boldsymbol{x} = \lambda(B + \epsilon I_d)\boldsymbol{x}$ instead. While this successfully gets rid of the singularity issue, it may create a more serious one in that the eventually computed top eigenvector likely falls into the null spaces of A and B and is thus useless for the underlying application.

The method is the so-called Locally Optimal Block Preconditioned Extended Conjugate Gradient method (LOBPECG) [Algorithm 2.3] [85] which combines LOBPCG of Knyazev [86] and the *inverse free Krylov subspace method* of Golub and Ye [87]. For our current application, we will simply use the version without preconditioning and blocking. Algorithm 2 outlines an adaption of [Algorithm 2.3] [85] for (A.1).

Algorithm 2: Locally Optimal Extended Conjugate Gradient method (LOECG) [85–87]

Input: eigenvalue problem (A.1), n_{kry}, tolerance *tol*;
Output: top eigenpair $(\lambda, \boldsymbol{x})$.
1: pick a random $\boldsymbol{x}_1 \in \mathbb{R}^{\mathrm{d}}$;
2: $\boldsymbol{x}_1 = B\boldsymbol{x}_1$, $\boldsymbol{x}_1 = \boldsymbol{x}_1/\|\boldsymbol{x}_1\|_2$, $\rho = \boldsymbol{x}_1^{\mathrm{T}}A\boldsymbol{x}_1/\boldsymbol{x}_1^{\mathrm{T}}B\boldsymbol{x}_1$;
3: $\boldsymbol{r} = A\boldsymbol{x}_1 - \rho B\boldsymbol{x}_1$, $\mathtt{res} = \|\boldsymbol{r}\|_2/(\|A\|_2 + |\rho|\|B\|_2)$;
4: $\boldsymbol{x}_0 = 0$;
5: **while** res $\geqslant$ *tol* **do**
6: compute an orthonormal basis matrix Z of the Krylov subspace

$$\mathscr{R}(Z) = \mathscr{R}\left([\boldsymbol{x}_1, (A-\rho B)\boldsymbol{x}_1, \ldots, (A-\rho B)^{n_{\mathrm{kry}}}\boldsymbol{x}_1]\right); \tag{A.2}$$

7: $\boldsymbol{p} = \boldsymbol{x}_0 - Z\left(Z^{\mathrm{T}}\boldsymbol{x}_0\right)$, $W = [Z, \boldsymbol{p}/\|\boldsymbol{p}\|_2]$;
8: compute the top eigenpair $(\rho, \boldsymbol{z})$ of $W^{\mathrm{T}}AW - \lambda W^{\mathrm{T}}BW$, where $\|\boldsymbol{z}\|_2 = 1$;
9: $\boldsymbol{x}_0 = \boldsymbol{x}_1$;
10: $\boldsymbol{x}_1 = W\boldsymbol{z}$, $\boldsymbol{r} = A\boldsymbol{x}_1 - \rho B\boldsymbol{x}_1$, $\mathtt{res} = \|\boldsymbol{r}\|_2/(\|A\|_2 + |\rho|\|B\|_2)$;
11: **end while**
12: **return** $(\rho, \boldsymbol{x}_1)$.

A few comments regarding this algorithm and its efficient implementation are in order:

1. There is no need to use $\|A\|_2$ and $\|B\|_2$ exactly. Some very rough estimates are just good enough so long as the estimates have the same magnitudes, respectively.
2. At line 2, it is to make sure $\boldsymbol{x}_1 \in \mathscr{R}(B)$.
3. There are two parameters to choose: the order n_{kry} of the Krylov space (A.2) and the stopping tolerance *tol*. There is no easy way to determine what the optimal n_{kry} is. In general, the larger n_{kry} is, the faster the convergence, but then more work in generating the orthonormal basis matrix Z. Usually $n_{\mathrm{kry}} = 10$ is good. For applications that required accuracy is not too stringent, $tol = 10^{-6}$ is often more than adequate.
4. The orthonormal basis matrix Z can be efficiently computed by the symmetric Lanczos process [88]. For better numerical stability in making sure $Z^{\mathrm{T}}Z = I$ within the working precision, re-orthogonalization may be necessary.
5. At line 7, some guard step must be taken. For example, in the first iteration $\boldsymbol{x}_0 = 0$ and so $\boldsymbol{p} = 0$. We should just let $W = Z$. In the subsequent iterations, we will have to test whether $\boldsymbol{x}_0$ is in or nearly in $\mathscr{R}(Z)$. For that purpose, we need another tolerance, e.g., if $\|\boldsymbol{p}\|_2 \leqslant 10^{-12}$, then we will regard already $\boldsymbol{x}_0 \in \mathscr{R}(Z)$ and set $W = Z$; otherwise, re-orthogonalize $\boldsymbol{p}$ against $Z : \boldsymbol{p} = \boldsymbol{p} - Z\left(Z^{\mathrm{T}}\boldsymbol{p}\right)$ to make sure $W^{\mathrm{T}}W = I$ within the working precision.
6. At line 8, AW and BW, except their last columns, are likely already computed at the time of generating Z at line 6. They should be reused here to save work.
7. The eigenvalue problem for $W^{\mathrm{T}}AW - \lambda W^{\mathrm{T}}BW$ is of very small size $(n_{\mathrm{kry}} + 1) \times (n_{\mathrm{kry}} + 1)$ at most and also $W^{\mathrm{T}}BW \succ 0$ as guaranteed by Lemma 1 below. It can be solved by first computing the Cholesky decomposition $W^{\mathrm{T}}BW = R^{\mathrm{T}}R$ and then the full eigen-decomposition of $R^{-\mathrm{T}}\left(W^{\mathrm{T}}AW\right)R^{-1}$. Finally, $\boldsymbol{z} = R^{-1}\boldsymbol{w}$, where $\boldsymbol{w}$ is the top eigenvector of $R^{-\mathrm{T}}\left(W^{\mathrm{T}}AW\right)R^{-1}$.

Lemma 1. In Algorithm 2, $\mathscr{R}(W) \subseteq \mathscr{R}(B)$ and thus $W^{\mathrm{T}}BW \succ 0$.

Proof. Initially, after line 2, $\boldsymbol{x}_1 \in \mathscr{R}(B)$. Therefore at (A.2), $\mathscr{R}(Z) \subseteq \mathscr{R}(B)$ because $\mathscr{R}(A) \subseteq \mathscr{R}(B)$. In the first iteration of the **while**-loop, $\boldsymbol{x}_0 = 0$ and $W = Z$ and so $\mathscr{R}(W) \subseteq \mathscr{R}(B), \boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathscr{R}(B)$. Inductively, each time at the beginning of executing the **while**-loop, we have $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathscr{R}(B)$. So we will have at line 7, $\boldsymbol{p} \in \mathscr{R}(B)$ and $\mathscr{R}(Z) \subseteq \mathscr{R}(B)$, implying $\mathscr{R}(W) \subseteq \mathscr{R}(B)$. Consequently, at the conclusion of executing the **while**-loop, we still have $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathscr{R}(B)$. Since $B \succeq 0$ and $\mathscr{R}(W) \subseteq \mathscr{R}(B), W^{\mathrm{T}}BW$ must be positive definite.

Appendix B. Proof of Theorem 1

We will show that the feasible sets for (34) and (35) are the same and $f(\{\boldsymbol{q}_s\}) = f_\ell(\{\boldsymbol{q}_s\})$ for any vector $\{\boldsymbol{q}_s\}$ in the feasible set.

Let $\{\boldsymbol{q}_s\}$ satisfy the constraints of (34). Since $\boldsymbol{q}_s^{\mathrm{T}} P_s^{(\ell)} = 0$, we have $\Pi_s^{(\ell)} \boldsymbol{q}_s = \boldsymbol{q}_s$. Since $\boldsymbol{q}_s \in \mathscr{R}(\Psi_{s,s}) = \mathscr{R}\left(\Psi_{s,s}^{1/2}\right)$ where $\Psi_{s,s}^{1/2}$ is the unique positive semi-definite square root of $\Psi_{s,s}$, we have $\boldsymbol{q}_s = \Psi_{s,s}^{1/2} \boldsymbol{w}_s$ for some $\boldsymbol{w}_s$. Therefore

$$\boldsymbol{q}_s = \Pi_s^{(\ell)} \boldsymbol{q}_s = \Pi_s^{(\ell)} \Psi_{s,s}^{1/2} \boldsymbol{w}_s \in \mathscr{R}\left(\Pi_s^{(\ell)} \Psi_{s,s}^{1/2}\right) = \mathscr{R}\left(\Pi_s^{(\ell)} \Psi_{s,s} \Pi_s^{(\ell)}\right),$$

$$\boldsymbol{q}_s^{\mathrm{T}} \Phi_{s,t} \boldsymbol{q}_t = \left[\Pi_s^{(\ell)} \boldsymbol{q}_s\right]^{\mathrm{T}} \Phi_{s,t} \left[\Pi_s^{(\ell)} \boldsymbol{q}_t\right] = \boldsymbol{q}_s^{\mathrm{T}} \Phi_{s,t}^{(\ell)} \boldsymbol{q}_t.$$

Hence $\{\boldsymbol{q}_s\}$ satisfies the constraints of (35) and $f(\{\boldsymbol{q}_s\}) = f_\ell(\{\boldsymbol{q}_s\})$. On the other hand, let $\{\boldsymbol{q}_s\}$ satisfy the constraints of (35). Since $\boldsymbol{q}_s \in \mathscr{R}\left(\Psi_{s,s}^{(\ell)}\right) = \mathscr{R}\left(\Pi_s^{(\ell)} \Psi_{s,s}^{1/2}\right)$, we have $\boldsymbol{q}_s = \Pi_s^{(\ell)} \Psi_{s,s}^{1/2} \boldsymbol{w}_s$ for some $\boldsymbol{w}_s$ and therefore

$$\boldsymbol{q}_s^{\mathrm{T}} P_s^{(\ell)} = \boldsymbol{w}_s^{\mathrm{T}} \Psi_{s,s}^{1/2} \Pi_s^{(\ell)} P_s^{(\ell)} = 0,$$

$$\boldsymbol{q}_s = \Psi_{s,s}^{1/2} \boldsymbol{w}_s - P_s^{(\ell)} \left[P_s^{(\ell)}\right]^{\mathrm{T}} \Psi_{s,s}^{1/2} \boldsymbol{w}_s \in \mathscr{R}\left(\Psi_{s,s}^{1/2}\right) = \mathscr{R}(\Psi_{s,s}).$$

That $\boldsymbol{q}_s^{\mathrm{T}} P_s^{(\ell)} = 0$ implies $\Pi_s^{(\ell)} \boldsymbol{q}_s = \boldsymbol{q}_s$ for all s, and therefore

$$\boldsymbol{q}_s^{\mathrm{T}} \Phi_{s,t} \boldsymbol{q}_t = \boldsymbol{q}_s^{\mathrm{T}} \Pi_s^{(\ell)} \Phi_{s,t} \Pi_t^{(\ell)} \boldsymbol{q}_t = \boldsymbol{q}_s^{\mathrm{T}} \Phi_{s,t}^{(\ell)} \boldsymbol{q}_t.$$

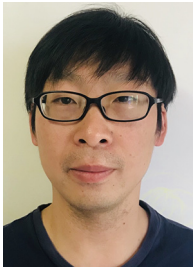
Hence also $\{\boldsymbol{q}_s\}$ satisfies the constraints of (34) and $f(\{\boldsymbol{q}_s\}) = f_\ell(\{\boldsymbol{q}_s\})$.

References

[1] X. Meng, H. Wang, L. Feng, The similarity-consensus regularized multi-view learning for dimension reduction, Knowledge-Based Systems 199 (2020) 105835.
[2] Q. Tian, C. Ma, M. Cao, S. Chen, H. Yin, A convex discriminant semantic correlation analysis for cross-view recognition, IEEE Transactions on Cybernetics (2020).
[3] X. You, J. Xu, W. Yuan, X.-Y. Jing, D. Tao, T. Zhang, Multi-view common component discriminant analysis for cross-view classification, Pattern Recognition 92 (2019) 37–51.
[4] X. Li, H. Zhang, R. Wang, F. Nie, Multi-view clustering: A scalable and parameter-free bipartite graph fusion method, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).
[5] J. Yin, S. Sun, Incomplete multi-view clustering with cosine similarity, Pattern Recognition 123 (2022) 108371.
[6] M. Yang, C. Deng, F. Nie, Adaptive-weighting discriminative regression for multi-view classification, Pattern Recognition 88 (2019) 236–245.
[7] S. Sun, D. Zong, Lcbm: A multi-view probabilistic model for multi-label classification, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).
[8] X. Zou, S. Wu, E.M. Bakker, X. Wang, Multi-label enhancement based self-supervised deep cross-modal hashing, Neurocomputing 467 (2022) 138–162.
[9] Z. Wang, L. Wang, H. Huang, Sparse additive discriminant canonical correlation analysis for multiple features fusion, Neurocomputing 463 (2021) 185–197.
[10] J. Zhang, L. Jing, J. Tan, Cross-regression for multi-view feature extraction, Knowledge-Based Systems 200 (2020) 105997.
[11] Y. Chen, S. Wang, C. Peng, G. Lu, Y. Zhou, Partial tubal nuclear norm regularized multi-view learning, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 1341–1349.
[12] Y. Xie, W. Zhang, Y. Qu, L. Dai, D. Tao, Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning, IEEE Transactions on Cybernetics 50 (2) (2018) 572–586.
[13] S. Li, W. Wang, W.-T. Li, P. Chen, Multi-view representation learning with manifold smoothness, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 8447–8454.
[14] Z. Huang, J.T. Zhou, H. Zhu, C. Zhang, J. Lv, X. Peng, Deep spectral representation learning from multi-view data, IEEE Transactions on Image Processing 30 (2021) 5352–5362.
[15] Y. Mao, X. Yan, Q. Guo, Y. Ye, Deep mutual information maximin for cross-modal clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 8893–8901.
[16] Y. Peng, J. Qi, CM-GANs: Cross-modal generative adversarial networks for common representation learning, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 15 (1) (2019) 1–24.
[17] S. Kaya, E. Vural, Learning multi-modal nonlinear embeddings: Performance bounds and an algorithm, IEEE Transactions on Image Processing 30 (2021) 4384–4394.
[18] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: Recent progress and new challenges, Information Fusion 38 (2017) 43–54.
[19] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, arXiv preprint arXiv:1304.5634 (2013).
[20] H. Hotelling, Relations between two sets of variates, Biometrika 28 (3–4) (1936) 321–377.
[21] A.A. Nielsen, Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data, IEEE Transactions on Image Processing 11 (3) (2002) 293–305.
[22] D.R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Computation 16 (12) (2004) 2639–2664.
[23] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correlation analysis, in: International Conference on Machine Learning, 2013, pp. 1247–1255.
[24] A. Sharma, A. Kumar, H. Daume, D.W. Jacobs, Generalized multiview analysis: A discriminative latent space, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2160–2167.
[25] G. Cao, A. Iosifidis, K. Chen, M. Gabbouj, Generalized multi-view embedding for visual recognition and cross-modal retrieval, IEEE Transactions on Cybernetics 48 (9) (2018) 2542–2555.
[26] S. Sun, X. Xie, M. Yang, Multiview uncorrelated discriminant analysis, IEEE Transactions on Cybernetics 46 (12) (2016) 3272–3284.
[27] A. Mandal, P. Maji, Faroc: fast and robust supervised canonical correlation analysis for multimodal omics data, IEEE Transactions on Cybernetics 48 (4) (2018) 1229–1241.
[28] M. Xu, Z. Zhu, X. Zhang, Y. Zhao, X. Li, Canonical correlation analysis with l2, 1-norm for multiview data representation, IEEE Transactions on Cybernetics 50 (11) (2020) 4772–4782.
[29] L. Sun, S. Ji, J. Ye, Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (1) (2010) 194–200.
[30] L. Wang, L.-H. Zhang, Z. Bai, R.-C. Li, Orthogonal canonical correlation analysis and applications, Optimization Methods and Software (2020) 1–21.
[31] X. Shen, Q. Sun, Y. Yuan, Orthogonal canonical correlation analysis and its application in feature fusion, in: Proceedings of the 16th International Conference on Information Fusion, 2013, pp. 151–157.
[32] J.P. Cunningham, Z. Ghahramani, Linear dimensionality reduction: Survey, insights, and generalizations, J. Mach. Learning Res. 16 (2015) 2859–2900.
[33] L. Zhang, L. Wang, Z. Bai, R.-C. Li, A self-consistent-field iteration for orthogonal cca, in: IEEE Transactions on Pattern Analysis and Machine IntelligenceTo appear, 2020.
[34] X. Shen, Q. Sun, Orthogonal multiset canonical correlation analysis based on fractional-order and its application in multiple feature extraction and recognition, Neural Processing Letters 42 (2) (2015) 301–316.
[35] L. Wang, R.-C. Li, A scalable algorithm for large-scale unsupervised multi-view partial least squares, IEEE Transactions on Big DataTo appear (2020).
[36] M.-L. Zhang, Z.-H. Zhou, Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038–2048.
[37] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst (editors), Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000.
[38] G.H. Golub, C.F. Van Loan, Matrix Computations, 4th Edition., Johns Hopkins University Press, Baltimore, Maryland, 2013.
[39] S. Guo, L. Feng, Z.-B. Feng, Y.-H. Li, Y. Wang, S.-L. Liu, H. Qiao, Multi-view laplacian least squares for human emotion recognition, Neurocomputing 370 (2019) 78–87.
[40] J. Chen, G. Wang, G.B. Giannakis, Graph multiview canonical correlation analysis, IEEE Transactions on Signal Processing 67 (11) (2019) 2826–2838.
[41] Y. Ito, T. Ogawa, M. Haseyama, Sfemcca: Supervised fractional-order embedding multiview canonical correlation analysis for video preference estimation, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 3086–3090.
[42] B. Jiang, J. Xiang, X. Wu, W. He, L. Hong, W. Sheng, Robust adaptive-weighting multi-view classification, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3117–3121.
[43] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, International Journal of Data Warehousing and Mining (IJDWM) 3 (3) (2007) 1–13.
[44] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineering 26 (8) (2013) 1819–1837.
[45] W. Liu, H. Wang, X. Shen, I. Tsang, The emerging trends of multi-label learning, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[46] C. Zhang, Z. Yu, Q. Hu, P. Zhu, X. Liu, X. Wang, Latent semantic aware multi-view multi-label classification, in: Thirty-second AAAI conference on artificial intelligence, 2018.
[47] Q. Tan, G. Yu, J. Wang, C. Domeniconi, X. Zhang, Individuality-and commonality-based multiview multilabel learning, IEEE Transactions on Cybernetics (2019).
[48] C. Zhu, D. Miao, Z. Wang, R. Zhou, L. Wei, X. Zhang, Global and local multi-view multi-label learning, Neurocomputing 371 (2020) 67–77.
[49] V. Ranjan, N. Rasiwasia, C. Jawahar, Multi-label cross-modal retrieval, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 4094–4102.
[50] X. Shu, G. Zhao, Scalable multi-label canonical correlation analysis for cross-modal retrieval, Pattern Recognition 115 (2021) 107905.
[51] K. Maeda, S. Takahashi, T. Ogawa, M. Haseyama, Multi-feature fusion based on supervised multi-view multi-label canonical correlation projection, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 3936–3940.
[52] F. Zhang, X. Jia, W. Li, Tensor-based multi-view label enhancement for multi-label learning, IJCAI (2020) 2369–2375.
[53] Z.-S. Chen, X. Wu, Q.-G. Chen, Y. Hu, M.-L. Zhang, Multi-view partial multi-label learning with graph-based disambiguation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 3553–3560.
[54] X. Li, S. Chen, A concise yet effective model for non-aligned incomplete multi-view and missing multi-label learning, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).
[55] Y. Zhang, J. Wu, Z. Cai, S.Y. Philip, Multi-view multi-label learning with sparse feature selection for image annotation, IEEE Transactions on Multimedia 22 (11) (2020) 2844–2857.
[56] I.T. Jolliffe, Principal components in regression analysis, in: Principal Component Analysis, Springer, 1986, pp. 129–155.
[57] J. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, Journal of Machine Learning Research 6 (Apr) (2005) 483–502.
[58] H. Wang, S. Yan, D. Xu, X. Tang, T. Huang, Trace ratio vs. ratio trace for dimensionality reduction, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.
[59] L.-H. Zhang, L.-Z. Liao, M.K. Ng, Fast algorithms for the generalized Foley-Sammon discriminant analysis, SIAM J. Matrix Anal. Appl. 31 (4) (2010) 1584–1605.
[60] E. Kokiopoulou, Y. Saad, Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2143–2156.
[61] D. Cai, X. He, Orthogonal locality preserving indexing, in: Proceedings of the 28th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2005, pp. 3–10.
[62] K.Q. Weinberger, F. Sha, L.K. Saul, Learning a kernel matrix for nonlinear dimensionality reduction, in: Proceedings of the Twenty-first International Conference on Machine Learning, 2004, p. 106.
[63] L. Wang, H. Yin, J. Zhang, Density-based distance preserving graph: Theoretical and practical analyses, IEEE Transactions on Neural Networks and Learning Systems (2021).
[64] L. Wang, R.-C. Li, Learning low-dimensional latent graph structures: A density estimation approach, IEEE Transactions on Neural Networks and Learning Systems 31 (4) (2019) 1098–1112.
[65] P. Hu, D. Peng, Y. Sang, Y. Xiang, Multi-view linear discriminant analysis network, IEEE Transactions on Image Processing 28 (11) (2019) 5352–5365.
[66] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization Algorithms On Matrix Manifolds, Princeton University Press, 2008.
[67] J. Nocedal, S. Wright, Numerical Optimization, 2nd Edition., Springer, 2006.
[68] Z. Wen, W. Yin, A feasible method for optimization with orthogonality constraints, Math. Program. 142 (1–2) (2013) 397–434.
[69] A. Gretton, O. Bousquet, A. Smola, B. Schölkopf, Measuring statistical dependence with hilbert-schmidt norms, in: International Conference on Algorithmic Learning Theory, Springer, 2005, pp. 63–77.
[70] M.T. Chu, J.L. Watterson, On a multivariate eigenvalue problem, part I: Algebraic theory and a power method, SIAM J. Sci. Comput. 14 (5) (1993) 1089–1106.
[71] L.-H. Zhang, Riemannian Newton method for the multivariate eigenvalue problem, SIAM J. Matrix Anal. Appl. 31 (5) (2010) 2972–2996.
[72] L.-H. Zhang, Riemannian trust-region method for the maximal correlation problem, Numer. Funct. Anal. Optim. 33 (3) (2012) 338–362.
[73] J. Wu, J.M. Rehg, Where am i: Place instance and category recognition using spatial pact, in: 2008 Ieee Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[74] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.
[75] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis & Machine Intelligence 7 (2002) 971–987.
[76] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Vol. 2, IEEE, 2006, pp. 2169–2178.
[77] F.-F. Li, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, Computer Vision and Image Understanding 106 (1) (2007) 59–70.
[78] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (11) (2008).
[79] P. Duygulu, K. Barnard, J.F. de Freitas, D.A. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: European Conference on Computer Vision, Springer, 2002, pp. 97–112.
[80] A. Makadia, V. Pavlovic, S. Kumar, A new baseline for image annotation, in: European Conference on Computer Vision, Springer, 2008, pp. 316–329.
[81] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2) (2010) 303–338.
[82] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 309–316.
[83] M. Guillaumin, J. Verbeek, C. Schmid, Multimodal semi-supervised learning for image classification, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 902–909.
[84] E. Gibaja, S. Ventura, A tutorial on multilabel learning, ACM Computing Surveys (CSUR) 47 (3) (2015) 1–38.
[85] R.-C. Li, Rayleigh quotient based optimization methods for eigenvalue problems, in: Z. Bai, W. Gao, Y. Su (Eds.), Matrix Functions and Matrix Equations, Vol. 19 of Series in Contemporary Applied Mathematics, World Scientific, Singapore, 2015, pp. 76–108, lecture summary for 2013 Gene Golub SIAM Summer School.
[86] A.V. Knyazev, Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method, SIAM J. Sci. Comput. 23 (2) (2001) 517–541.
[87] G. Golub, Q. Ye, An inverse free preconditioned Krylov subspace methods for symmetric eigenvalue problems, SIAM J. Sci. Comput. 24 (2002) 312–334.
[88] J. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, PA, 1997.

Li Wang is an assistant professor with Department of Mathematics and Department of Computer Science and Engineering, University of Texas at Arlington, Texas, USA. She was a research assistant professor with Department of Mathematics, Statistics, and Computer Science at University of Illinois at Chicago, Chicago, USA from 2015 to 2017, and a postdoctoral fellow at University of Victoria, BC, Canada in 2015 and Brown University, USA, in 2014. She received her PhD from Department of Mathematics at University of California, San Diego, USA, in 2014, and MS from Xi'an Jiaotong University, Shaanxi, China in 2009 and BS in Information and Computing Science from China University of Mining and Technology, Jiangsu, China in 2006. Her research interests include large scale optimization, polynomial optimization and machine learning.



Lei-Hong Zhang is with School of Mathematical Sciences and Institute of Computational Science, Soochow University, Suzhou 215006, Jiangsu, China. He received his BS and MS degrees from Southeast University, China, in 2002 and 2005, respectively, and PhD from the Hong Kong Baptist University, China in 2008. His research interests include optimization, numerical linear algebra and machine learning.

Chungen Shen is currently an associate professor with the College of Science, University of Shanghai for Science and Technology, Shanghai, China. He received the BS degrees in applied mathematics from Anhui Normal University, People's Republic of China, in 2002, and received the MS and PhD degrees in operations research from Tongji University, People's Republic of China, in 2005 and 2010, respectively. His research interest includes numerical optimization and machine learning.

Ren-Cang Li is a professor with the Department of Mathematics, University of Texas at Arlington, Texas, USA. He received his BS from Xiamen University, China, in 1985, MS from the Chinese Academy of Science in 1988, and PhD from University of California, Berkeley, in 1995. He was awarded the 1995 Householder Fellowship in Scientific Computing by Oak Ridge National Laboratory, a Friedman memorial prize in Applied Mathematics from the University of California at Berkeley in 1996, and CAREER award from NSF in 1999. His research interest includes floating-point support for scientific computing, large and sparse linear systems, eigenvalue problems, and model reduction, machine learning, and unconventional schemes for differential equations.