A Scalable Algorithm for Large-scale Unsupervised Multi-view Partial Least Squares

Li Wang and Ren-cang Li

Abstract—We present an unsupervised multi-view partial least squares (PLS) by learning a common latent space from given multi-view data. Although PLS is a frequently used technique for analyzing relationships between two datasets, its extension to more than two views in unsupervised setting is seldom studied. In this paper, we fill up the gap, and our model bears similarity to the extension of canonical correlation analysis (CCA) to more than two sets of variables and is built on the findings from analyzing PLS, CCA and its variants. The resulting problem involves a set of orthogonality constraints on view-specific projection matrices, and is numerically challenging to existing methods that may have numerical instabilities and offers no orthogonality guarantee on view-specific projection matrices. To solve this problem, we propose a stable deflation algorithm that relies on proven numerical linear algebra techniques, can guarantee the orthogonality constraints, and simultaneously maximizes the covariance in the common space. We further adapt our algorithm to efficiently handle large-scale high-dimensional data. Extensive experiments have been conducted to evaluate the algorithm through performing two learning tasks, cross-modal retrieval and multi-view feature extraction. The results demonstrate that the proposed algorithm outperforms the baselines and is scalable for large-scale high-dimensional datasets.

Index Terms—Partial least squares, unsupervised subspace learning, multi-view learning

1 Introduction

In real-world applications, data is often collected in multiple views. They are the same object but from different perspectives. Multi-view data provides more information, but, at the same time, it creates difficulties due to large discrepancies among views. In the cross-modal retrieval [1], [2], it is needed to perform classification and retrieval on the gallery and query data in text and image that represent different views. These tasks are challenging since text and image are two heterogeneous concepts (views) from different feature spaces and there generally lacks a meaningful priori to directly compare two heterogeneous views. Multi-view learning [3], [4], [5] is designed to overcome such challenges by exploiting the consensual and complementary information among different views. A popular and natural approach [6], [7] is to first learn view-specific projections for all views – one projection for one view – and then project the original multi-view data from different views onto a common space by the view-specific projections to make comparison possible.

Several multi-view learning approaches have been studied in the literature (see survey papers [3], [4], [5] for details). Among them, subspace learning methods have been extensively explored and successfully applied to various learning scenarios, such as sparse low-rank approximation for incomplete data [8], online method for streaming multi-view data [9], adaptive graph learning for multi-view clustering and semi-supervised learning [10], and multi-view data representation learning for supervised learning [11] and sparse learning [12], and deep representation learning for multi-view data [13]. In this paper, we seek to study multi-view learning through an approach of unsupervised subspace learning using multivariate analysis techniques. Specifically, we assume that all views are generated from a common latent space and no label information is available during learning process.

Common unsupervised subspace learning approaches for two views include the classical canonical correlation analysis (CCA) [6] and partial least squares (PLS) [7]. CCA attempts to learn view-specific projections by maximizing the correlation between two views. It has been the workhorse for learning a common latent space, as evidenced by its successful applications in various domains [14]. CCA does not produce orthonormal projection matrices in the first place, but orthogonality is a preferred property for data visualization and metric preservation [15], [16], [17]. Due to the special structure of CCA, the orthogonality can be obtained by whitening the original view data as a preprocessing step [18]. Unfortunately, the covariance of the original views is no longer preserved after the whitening. Orthogonal CCA (OCCA) is proposed to solve the above issue by maximizing correlation between two views and simultaneously imposing orthogonality constraints on each individual view-specific projection matrices [18], [19], [20], [21]. The merit of OCCA compared to CCA is that the orthogonal projection matrices are obtained to maximize correlation while the original covariances are preserved. PLS [7] maximizes the covariance of two views in the common space based on the assumption that information is over-represented so they can be reduced. Some appealing properties brought by PLS are its abilities to handle datasets in which the numbers of features are more than the number of samples and there are massive colinearities between two sets of variables [22]. PLS has been successfully used in

- *Li Wang is with Department of Mathematics and Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019-0408, USA. Supported in part by NSF DMS-2009689. Email: li.wang@uta.edu.*
- *Ren-cang Li is with Department of Mathematics, University of Texas at Arlington, Arlington, TX 76019-0408, USA. Supported in part by NSF DMS-1719620 and DMS-2009689. Email: rcli@uta.edu.*

many applications such as cross-modal retrieval and pose estimation of face images [23]. An alternative interpretation of PLS is that it simultaneously maximizes the correlation and view-specific variances [24]. This explains the connections of PLS to CCA and OCCA: they are based on the same criterion of maximizing correlation but with different constraints on variances or covariance of the original data.

Due to the modeling variability for data with more than two views, various formulations can be used. The extension of CCA to more than two views, called the multiset CCA (MCCA), has been extensively studied with various learning criteria, including maximizing the sum of correlations [25], maximizing the sum of the squared correlations, minimizing the smallest eigenvalue or the determinant of the correlation matrix [26], maximizing the sum of all the pairwise correlations and the high-order correlation [27], and exploiting Hessian for intrinsic local geometry [28]. In [29], twenty variants are proposed based on four types of constraints and five different objective functions. An OCCA extension to more than two views has also been explored [30]. However, the extension of PLS to more than two views is seldom studied. We notice that the recent work [24] explored PLS for more than two views for the multivariate regression problem, in which the regressors are represented in terms of multiple views and the response is required so that the covariance between a linear combination of regressors and the response in the common space is maximized. Hence, the work [24] targets at a regression model and is different from the study of this work for unsupervised subspace learning.

To fill up the gap of PLS for handling more than two views in unsupervised learning, we will first explain a connection of PLS to CCA and OCCA, and then propose our Unsupervised Multi-view Partial Least Squares (UMvPLS), a multi-view variant of PLS. It bears similarity to the extensions of CCA and OCCA to their multi-view versions. The resulting problem is numerically challenging due to the set of orthogonality constraints. To solve the problem, we propose a stable deflation algorithm so that the orthogonality constraints are automatically satisfied while simultaneously the covariance in the common space is being maximized. We notice that existing methods in [19], [24], [30] cannot guarantee orthonormality of computed projection matrices and can even breakdown, more often than one might think, due to their numerical instability. The proposed algorithm resolves all these issues. Moreover, we adapt our algorithm to efficiently handle large-scale high-dimensional data. Extensive experiments have been conducted to evaluate our algorithm while conducting two learning tasks: cross-modal retrieval and multi-view feature extraction. The experimental results show that our proposed method outperforms CCA, OCCA and their multi-view extensions, especially when the common space has a small dimension, and runs much faster than CCA methods for large-scale high-dimensional data.

The rest of this paper is organized as follows. We first review the existing methods that are most related to this work in Section 2. In Section 3, we present an extension of PLS for unsupervised multi-view subspace learning and propose novel optimization algorithms for large-scale high-dimensional data. Extensive experiments are conducted in Section 4. Finally, we draw our conclusions in Section 5.

Notation. $\mathbb{R}^{m\times n}$ is the set of $m\times n$ real matrices, $\mathbb{R}^n = \mathbb{R}^{n\times 1}$, and $\mathbb{O}^{n\times k} = \{X \in \mathbb{R}^{n\times k} : X^{\mathrm{T}}X = I_k\}$, where $I_k \in \mathbb{R}^{k\times k}$ is the identity matrix. $\mathbf{1}_n \in \mathbb{R}^n$ is the vector of all ones. $\|\boldsymbol{x}\|_2$ is the 2-norm of vector $\boldsymbol{x} \in \mathbb{R}^n$. For $B \in \mathbb{R}^{m\times n}$, $\mathcal{R}(B)$ is the column space and, when B is square, $\mathrm{tr}(B)$ is the trace of B.

2 Related Work

We review some existing methods such as CCA, OCCA, and PLS for unsupervised multi-view learning. Denote the ℓ datasets associated with the multiple views, in terms of data matrices, by

$$S_i \in \mathbb{R}^{n_i\times q} \quad \text{for } i = 1, 2, \ldots, \ell, \tag{1}$$

where n_i is the number of features of the ith view and q is the number of samples. Without loss of generality, assume that each dataset is centered, i.e., $S_i\mathbf{1}_q = 0$; otherwise they can be preprocessed by

$$S_i \leftarrow S_i\left(I_q - \frac{1}{q}\mathbf{1}_q\mathbf{1}_q^{\mathrm{T}}\right) = S_i - \frac{1}{q}(S_i\mathbf{1}_q)\mathbf{1}_q^{\mathrm{T}} \quad \forall i. \tag{2}$$

2.1 CCA

CCA is a two-view multivariate statistical method [6], where the variables of observations can be partitioned into two sets, leading to the two views of the data, S_1 and S_2, and its goal is to find a common subspace so that two views are maximally correlated within in the subspace. Let $Y_i \in \mathbb{R}^{n_i\times k}, \forall i = 1, 2$, be the projection matrices and $X_i = Y_i^{\mathrm{T}}S_i \in \mathbb{R}^{k\times n_i}$ be the embedded points in the common subspace. The correlation between two views in the common subspace is naturally defined as

$$\frac{\mathrm{tr}(X_1X_2^{\mathrm{T}})}{\|X_1\|_{\mathrm{F}}\|X_2\|_{\mathrm{F}}}, \tag{3}$$

where $\|\cdot\|_{\mathrm{F}}$ is the matrix Frobenius norm. By maximizing the correlation (3), the optimization problem of the CCA model is formulated as

$$\max_{Y_1,Y_2}\left\{\rho(\{Y_i\}) := \frac{\mathrm{tr}(Y_1^{\mathrm{T}}S_1S_2^{\mathrm{T}}Y_2)}{\sqrt{\mathrm{tr}(Y_1^{\mathrm{T}}S_1S_1^{\mathrm{T}}Y_1)\,\mathrm{tr}(Y_2^{\mathrm{T}}S_2S_2^{\mathrm{T}}Y_2)}}\right\}. \tag{4}$$

Problem (4) can be solved by the singular value decomposition (SVD) [31]. Specifically, let

$$Y_1 = (S_1S_1^{\mathrm{T}})^{-\frac{1}{2}}\Psi_1, Y_2 = (S_2S_2^{\mathrm{T}})^{-\frac{1}{2}}\Psi_2. \tag{5}$$

Problem (4) can be equivalently reformulated as the SVD problem:

$$\max_{\Psi_1,\Psi_2}\ \mathrm{tr}(\Psi_1^{\mathrm{T}}W\Psi_2)\ \text{ s.t. }\ \Psi_1^{\mathrm{T}}\Psi_1 = I_k, \Psi_2^{\mathrm{T}}\Psi_2 = I_k, \tag{6}$$

where $W = (S_1S_1^{\mathrm{T}})^{-\frac{1}{2}}S_1S_2^{\mathrm{T}}(S_2S_2^{\mathrm{T}})^{-\frac{1}{2}}$. The optimal Ψ_1 and Ψ_2 are the left and right singular vector matrices of W corresponding to its top k singular values [32, p.195]. After (6) is solved, the optimal solutions Y_1 and Y_2 can be recovered according to (5).

2.2 OCCA

According to (5) and (6), it is clear that the columns of Ψ_1 and Ψ_2 forms two orthonormal bases, but the columns of Y_1 and Y_2 usually do not unless data points as the columns of both S_1 and S_2 are orthogonal, respectively. However, this case seldom happens in the real world. Another interpretation is that the classical CCA whitens dataset matrices S_1 and S_2, and then orthogonally projects these whitened data into a common space such that correlation is maximized [18]. This whitening step causes the change of correlation of the original data. To overcome this issue, an OCCA [18] is proposed to maximize the correlation of two views in the common space with orthogonality constraints:

$$\max_{Y_1,Y_2} \rho(\{Y_i\}) \text{ s.t. } Y_1^{\mathrm{T}}Y_1 = I_k, Y_2^{\mathrm{T}}Y_2 = I_k. \tag{7}$$

In (7), the original covariance of two views is maintained. Problem (7) is no longer solvable by SVD, unlike CCA. Several methods were proposed to solve this problem, including generic optimization methods on matrix manifolds [18], and a method (which we will call OCCA-SSY) in [19] via deflation that somewhat solves (7).

2.3 PLS

PLS was originally developed as a method for supervised multivariate analysis [7]. It aims to find orthonormal bases so that the variances and correlation are all maximized in a common space. Specifically, PLS is to solve the following problem

$$\begin{aligned}\max_{Y_1,Y_2} \rho(\{Y_i\})\sqrt{\mathrm{tr}(Y_1^{\mathrm{T}}S_1S_1^{\mathrm{T}}Y_1)\,\mathrm{tr}(Y_2^{\mathrm{T}}S_2S_2^{\mathrm{T}}Y_2)} \\ \text{s.t. } Y_1^{\mathrm{T}}Y_1 = I_k, Y_2^{\mathrm{T}}Y_2 = I_k.\end{aligned} \tag{8}$$

According to (3), the objective function of (8) is the same as the covariance matrix between the projected input and output data, that is, $\mathrm{tr}(Y_1^{\mathrm{T}}S_1S_2^{\mathrm{T}}Y_2)$. In other words, PLS is equivalent to the following problem

$$\max_{Y_1,Y_2} \mathrm{tr}(Y_1^{\mathrm{T}}S_1S_2^{\mathrm{T}}Y_2) \text{ s.t. } Y_1^{\mathrm{T}}Y_1 = I_k, Y_2^{\mathrm{T}}Y_2 = I_k. \tag{9}$$

Mathematically, this is the same as (6), explicitly solvable by the SVD of $S_1S_2^{\mathrm{T}}$.

In what follows, we consider PLS as an unsupervised subspace learning method since two views are equivalently treated with no distinction for regressors and responses as commonly used in regression models. In this case, PLS not only maximizes correlation and variances, but also directly produces the orthonormal projection matrices, in contrast to CCA.

2.4 CCA and OCCA for multiple sets of variables

Multiset CCA (MCCA) is the extension of CCA for mult-variables. Different from two views, multi-view data leads to modeling flexibility for MCCA. Here, we briefly introduce one widely used variant by seeking projections to maximize the sum of the pairwise correlations between any two canonical variates given by

$$\max_{(Y_1,\ldots,Y_\ell)\in\mathcal{Y}} \sum_{i,j=1}^{\ell} \mathrm{tr}(Y_i^{\mathrm{T}}S_iS_j^{\mathrm{T}}Y_j), \tag{10}$$

where $\mathcal{Y}$ is some feasible set of projection matrices. Two commonly used ones are [29]:

$$\mathcal{Y} := \Big\{(Y_1,\ldots,Y_\ell) : \sum_{i=1}^{\ell} Y_i^{\mathrm{T}}S_iS_i^{\mathrm{T}}Y_i = I_k\Big\}; \tag{11}$$

$$\mathcal{Y} := \Big\{(Y_1,\ldots,Y_\ell) : Y_i^{\mathrm{T}}S_iS_i^{\mathrm{T}}Y_i = I_k, \forall i = 1,\ldots,\ell\Big\}. \tag{12}$$

MCCA (10) with condition (11) can be turned into a generalized eigenvalue problem [29].

Similarly, OCCA is extended for multi-variables in [30] and the extension does not directly solve (10). For the ease of reference, we name the extension as OMCCA-SS [30].

3 Unsupervised Multi-view PLS

3.1 Motivation and contributions

PLS has the advantages of having the variances maximized and producing orthonormal projection matrices in contrast to CCA, and having the covariance maximized instead of maintaining covariance as modeled by OCCA [18]. However, the extension of PLS to multiset variables is seldom studied. We notice that in the most recent work [24] it is proposed to extend supervised PLS for multi-view learning where the relationships between the response and the weighted combination of multi-view regressors are modeled. In addition, the optimization method proposed in [24] cannot guarantee that projection matrices are orthonormal and also its convergence analysis is questionable.

In what follows, we will formulate a multi-view PLS for unsupervised subspace learning – UMvPLS, and propose two algorithms for its robust and efficient implementations with Algorithm 1 geared towards modest-scale multi-view data while Algorithm 2 towards large-scale multi-view data. Because they are built upon well-developed numerical linear algebra techniques, both algorithms are free from any numerical instability issues previously witnessed for OCCA-SSY [19] and OMCCA-SS [30], and they can provably guarantee that computed projection matrices $\{Y_i\}$ are orthonormal.

3.2 Formulation of unsupervised Multi-view PLS

Let $n = \sum_{i=1}^{\ell} n_i$. By concatenating all ℓ data matrices vertically, we have the following compact representation:

$$\mathscr{S} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_\ell \end{bmatrix} \in \mathbb{R}^{n\times q}, \mathscr{S}\mathbf{1}_q = \begin{bmatrix} S_1\mathbf{1}_q \\ S_2\mathbf{1}_q \\ \vdots \\ S_\ell\mathbf{1}_q \end{bmatrix} = 0. \tag{13}$$

We further partition the covariance matrix $\mathscr{S}\mathscr{S}^{\mathrm{T}}$ and projection matrix Y as

$$\mathscr{S}\mathscr{S}^{\mathrm{T}} = \mathscr{A} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1\ell} \\ C_{21} & C_{22} & \cdots & C_{2\ell} \\ \vdots & \vdots & \ddots & \vdots \\ C_{\ell 1} & C_{\ell 2} & \cdots & C_{\ell\ell} \end{bmatrix}, Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\ell \end{bmatrix}, \tag{14}$$

where $C_{ij} = S_iS_j^{\mathrm{T}}$ and $Y_i \in \mathbb{R}^{n_i\times k}, \forall i,j = 1,\ldots,\ell$.

Following the modeling process of MCCA, we propose our unsupervised multi-view PLS (UMvPLS) model as the following optimization problem

$$\begin{aligned}\max_{\{Y_i\}} \Big\{f(\{Y_i\}) := \mathrm{tr}(Y^{\mathrm{T}}\mathscr{A}Y)\Big\} \\ \text{s.t. } Y_i^{\mathrm{T}}Y_i = I_k, \forall i = 1,\ldots,\ell.\end{aligned} \tag{15}$$

The difference between (15) and OMCCA-SS [30] is in their feasible sets of projections, analogously to that between PLS and OCCA as discussed in Section 2. As a result, problem (15) is also not an easy optimization problem to solve. Its KKT condition is given by a multi-parameter eigenvalue problem

$$\mathscr{A}\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\ell \end{bmatrix} = \begin{bmatrix} Y_1\Lambda_1 \\ Y_2\Lambda_2 \\ \vdots \\ Y_\ell\Lambda_\ell \end{bmatrix}, \tag{16}$$

where $\Lambda_i^{\mathrm{T}} = \Lambda_i \in \mathbb{R}^{k\times k}$ for $1 \le i \le \ell$. By (16), we have for (15)

$$f(\{Y_i\}) = \sum_i \mathrm{tr}(\Lambda_i). \tag{17}$$

It is not clear how to find the solution to (16) that maximizes $\sum_i \mathrm{tr}(\Lambda_i)$. The multi-parameter eigenvalue problem (16) is numerically challenging and there is no existing numerical linear algebra technique that can readily solve it, even for the case $k = 1$. For this reason, in what follows, we will propose efficient algorithms to approximately solve (15).

3.3 An incremental algorithm for UMvPLS

Due to the difficulty in solving the multi-parameter eigenvalue problem (16), we seek to compute one column of Y at a time with the help of a deflation idea that traces back to [33] so that the ℓ orthogonality constraints in (15) are satisfied, $\mathcal{R}(Y_i) \subset \mathcal{R}(S_i)\ \forall i$ (thus range constrained), and at the same time $f(\{Y_i\})$ is decently maximized. The approach is incremental in nature and we shall call it an incremental algorithm for UMvPLS.

The building block of our overall algorithm can be best described by the case $k = 1$ of (15). Instead of solving (15) directly, we consider a simpler problem

$$\max_{\{z_i\}} f(\{z_i\}) \equiv \max_{\boldsymbol{z}} \boldsymbol{z}^{\mathrm{T}}\mathscr{A}\boldsymbol{z} \text{ s.t. } \sum_i z_i^{\mathrm{T}} z_i \equiv \boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} = 1, \tag{18}$$

where $\boldsymbol{z}^{\mathrm{T}} = [z_1^{\mathrm{T}}, z_2^{\mathrm{T}}, \ldots, z_\ell^{\mathrm{T}}]^{\mathrm{T}}$. This problem is equivalent to the standard eigenvalue problem

$$\mathscr{A}\boldsymbol{z} = \lambda\boldsymbol{z}. \tag{19}$$

Let $\lambda_{\max}(\mathscr{A})$ be the largest eigenvalue of $\mathscr{A}$ and $\boldsymbol{z}^{\mathrm{opt}}$ the corresponding unit eigenvector. Then $\boldsymbol{z}^{\mathrm{opt}}$ is a maximizer of (18) and $f(\boldsymbol{z}^{\mathrm{opt}}) = \lambda_{\max}(\mathscr{A})$. It is worth noting that problem (18) is not equivalent to problem (15) with $k = 1$. The key part of the proposed algorithm is to transform the solution $\boldsymbol{z}^{\mathrm{opt}}$ to the simpler problem (18) to an approximate solution $\boldsymbol{y}^{\mathrm{opt}}$ to problem (15) for $k = 1$ as follows: first partition $\boldsymbol{z}^{\mathrm{opt}}$ conformally as

$$\boldsymbol{z}^{\mathrm{opt}} = \begin{bmatrix} z_1^{\mathrm{opt}} \\ z_2^{\mathrm{opt}} \\ \vdots \\ z_\ell^{\mathrm{opt}} \end{bmatrix} \quad \text{with } z_i^{\mathrm{opt}} \in \mathbb{R}^{n_i},$$

and let

$$\gamma_i = \|z_i^{\mathrm{opt}}\|_2,\ y_i^{\mathrm{opt}} = z_i^{\mathrm{opt}}/\gamma_i,\ \boldsymbol{y}^{\mathrm{opt}} = \begin{bmatrix} y_1^{\mathrm{opt}} \\ y_2^{\mathrm{opt}} \\ \vdots \\ y_\ell^{\mathrm{opt}} \end{bmatrix}. \tag{20}$$

Evidently, $\sum_i \gamma_i^2 = 1$. This $\boldsymbol{y}^{\mathrm{opt}}$ will be regarded as an approximate maximizer of (15) for the case $k = 1$. Since

$$\lambda_{\max}(\mathscr{A})z_i^{\mathrm{opt}} = \sum_j C_{ij}z_j^{\mathrm{opt}} = S_i\sum_j S_j^{\mathrm{T}}z_j^{\mathrm{opt}} \in \mathcal{R}(S_i), \tag{21}$$

we conclude that $y_i^{\mathrm{opt}} = z_i^{\mathrm{opt}}/\gamma_i \in \mathcal{R}(S_i)\ \forall i$.

In terms of the singular value decomposition (SVD), $\boldsymbol{z}^{\mathrm{opt}}$ is also the top left singular vector of $\mathscr{S}$ associated with the largest singular value $\sigma_{\max}(\mathscr{S})$ of $\mathscr{S}$, i.e., $\|\mathscr{S}^{\mathrm{T}}\boldsymbol{z}^{\mathrm{opt}}\|_2 = \sigma_{\max}(\mathscr{S})$. In fact,

$$\mathscr{A} = \mathscr{S}\mathscr{S}^{\mathrm{T}},\ \lambda_{\max}(\mathscr{A}) = [\sigma_{\max}(\mathscr{S})]^2.$$

From the numerical point of view, computing $\boldsymbol{z}^{\mathrm{opt}}$ via calculating the top left singular vector of $\mathscr{S}$ is both more accurate and economical (especially when $\sum_i n_i \gg q$ or $\ll q$) [34], [35]. Hence, the proposed method is extremely helpful for solving UMvPLS for large-scale data or high-dimension data in cases when either there are a large number of views or there some views residing in very high-dimensional spaces.

The set $\{y_i^{\mathrm{opt}}\}$ well represents the first set of most correlated unit vectors for the ℓ datasets, and it gives the first column of Y. For the next set of unit vectors representing the second most significant ones, each of which is orthogonal to the corresponding one in the set just computed, we propose to use a classical deflation idea [33] from numerical linear algebra:

1) update each S_i by
$$S_i \leftarrow \big[I_{n_i} - y_i^{\mathrm{opt}}(y_i^{\mathrm{opt}})^{\mathrm{T}}\big]S_i = S_i - y_i^{\mathrm{opt}}\big[(y_i^{\mathrm{opt}})^{\mathrm{T}}S_i\big]; \tag{22}$$
2) use these updated S_i to form $\mathscr{S}$ as in (13) and compute its top left singular vector $\hat{\boldsymbol{z}}^{\mathrm{opt}} = [\hat{z}_i^{\mathrm{opt}}]$;
3) post-process $\hat{\boldsymbol{z}}^{\mathrm{opt}}$ like in (20) to yield $\hat{\boldsymbol{y}}^{\mathrm{opt}} = [\hat{y}_i^{\mathrm{opt}}]$ for the next column of Y.

We claim that the orthogonality between each $\hat{y}_i^{\mathrm{opt}}$ and the corresponding y_i^{opt} is guaranteed. To see it, we let $\widehat{S}_i$ denote the updated S_i in (22), i.e., $\widehat{S}_i = S_i - y_i^{\mathrm{opt}}\big[(y_i^{\mathrm{opt}})^{\mathrm{T}}S_i\big]$. It can be verified that

$$(y_i^{\mathrm{opt}})^{\mathrm{T}}\widehat{S}_i = (y_i^{\mathrm{opt}})^{\mathrm{T}}\big[I_{n_i} - y_i^{\mathrm{opt}}(y_i^{\mathrm{opt}})^{\mathrm{T}}\big]S_i = 0.$$

By the range constraining property in (21), $\hat{y}_i^{\mathrm{opt}} = \widehat{S}_i w_i$ for some vector w_i and consequently

$$(y_i^{\mathrm{opt}})^{\mathrm{T}}\hat{y}_i^{\mathrm{opt}} = (y_i^{\mathrm{opt}})^{\mathrm{T}}\widehat{S}_i w_i = 0.$$

We summarize our algorithm for UMvPLS in Algorithm 1.

The brackets in the last expression of (22), as well as these at line 4 of Algorithm 1, shall be respected for numerical efficiency. Algorithm 1 as stated is suitable for a small to medium scale UMvPLS problem. Later in section 3.4, we will explain how it can be adapted for large scale multi-view data.

Theorem 1 below shows that the solution returned by Algorithm 1 satisfies the orthogonality constraints in problem (15).

Theorem 1. *Let $\{Y_i\}$ be the output of Algorithm 1. Then $Y_i^{\mathrm{T}}Y_i = I_k$ and $\mathcal{R}(Y_i) \subset \mathcal{R}(S_i)\ \forall i$.*

Proof. Denote the jth updated S_i at line 4 of Algorithm 1 by $S_i^{(j)}$, and $S_i^{(0)} = S_i$ is the input one. Similarly,

Algorithm 1 UMvPLS: an incremental algorithm

Input: $\{S_i \in \mathbb{R}^{n_i \times q}\}$ (each S_i is centered), integer $1 \le k \le \min\{n_1, \ldots, n_\ell, q\}$;
Output: $\{Y_i \in \mathbb{O}^{n_i \times k}\}$, the set of most correlated orthonormal projection matrices.
1: compute the most dominant left singular vector $\boldsymbol{z} = [z_1^{\mathrm{T}}, \ldots, z_\ell^{\mathrm{T}}]^{\mathrm{T}}$ of $\mathscr{S}$ in (13), where $z_i \in \mathbb{R}^{n_i}$;
2: $y_i^{(1)} = z_i/\|z_i\|_2$ for $i = 1, 2, \ldots, \ell$;
3: **for** $j = 1, 2 \ldots, k-1$ **do**
4: update $S_i \leftarrow S_i - y_i^{(j)} \left([y_i^{(j)}]^{\mathrm{T}} S_i \right)$ for $i = 1, 2, \ldots, \ell$;
5: compute the most dominant left singular vector $\boldsymbol{z} = [z_1^{\mathrm{T}}, \ldots, z_\ell^{\mathrm{T}}]^{\mathrm{T}}$ of $\mathscr{S}$ in (13) with the updated S_i, where $z_i \in \mathbb{R}^{n_i}$;
6: $y_i^{(j+1)} = z_i/\|z_i\|_2$ for $i = 1, 2, \ldots, \ell$;
7: **end for**
8: $Y_i = [y_i^{(1)}, \ldots, y_i^{(k)}]$ for $i = 1, 2, \ldots, \ell$;
9: **return** $\{Y_i \in \mathbb{O}^{n_i \times k}\}$.

$$Y_i^{(j)} = [y_i^{(1)}, \ldots, y_i^{(j)}]. \tag{23}$$

We will prove

$$(Y_i^{(j)})^{\mathrm{T}} Y_i^{(j)} = I_j, \ \mathcal{R}(Y_i^{(j)}) \subset \mathcal{R}(S_i) \ \forall i \tag{24}$$

by induction on j. Evidently, the claims in (24) hold for $j = 1$, based on our discussions leading to Algorithm 1 in this section. Suppose they are true for $j \le t$. We have to prove them for $j = t+1$. To this end, we note

$$S_i^{(t)} = \prod_{j=1}^{t} \left(I_{n_i} - y_i^{(j)} [y_i^{(j)}]^{\mathrm{T}} \right) S_i = \left[I_{n_i} - Y_i^{(t)} (Y_i^{(t)})^{\mathrm{T}} \right] S_i$$

because $(Y_i^{(t)})^{\mathrm{T}} Y_i^{(t)} = I_t$. Immediately,

$$\mathcal{R}(S_i^{(t)}) \subset \mathcal{R}(S_i) + \mathcal{R}(Y_i^{(t)}) \subset \mathcal{R}(S_i).$$

By construction, $y_i^{(t+1)} \in \mathcal{R}(S_i^{(t)})$ and thus $y_i^{(t+1)} = S_i^{(t)} w_i$ for some vector w_i. Therefore,

$$\begin{aligned} (Y_i^{(t)})^{\mathrm{T}} y_i^{(t+1)} &= (Y_i^{(t)})^{\mathrm{T}} S_i^{(t)} w_i \\ &= (Y_i^{(t)})^{\mathrm{T}} \left[I_{n_i} - Y_i^{(t)} (Y_i^{(t)})^{\mathrm{T}} \right] S_i w_i = 0, \end{aligned}$$

leading to (24) for $j = t+1$. □

3.4 Scalable algorithm for UMvPLS on large-scale data

Algorithm 1 is not scalable for large-scale data since all S_i there are likely dense and computing a full SVD is an expensive operation. At lines 1 and 5 of Algorithm 1, if a full dense SVD is computed by, e.g., MATLAB's `svd` which is based on LAPACK [36], as a way to extract the top left singular vector, the cost is $O(\min\{nq^2, n^2q\})$ flops, where $n = \sum_i n_i$. Fortunately, only the top left singular vectors are required at both lines. Therefore a Krylov subspace type iterative method based on the Golub-Kahan bidiagonlization [35], [37], [38] can get the job done in $O(nq)$ flops for each top left singular vector. The saving will be even greater when original data matrices S_i (unlikely centered however) are sparse.

A key requirement of these Krylov subspace methods [35, chapter 10] is the ability to compute matrix-vector products, in our case,

Algorithm 2 UMvPLS: an incremental algorithm (scalable version)

Input: $\{S_i^{\mathrm{raw}} \in \mathbb{R}^{n_i \times q}\}$ (each S_i^{raw} is not necessarily centered), integer $1 \le k \le \min\{n_1, \ldots, n_\ell, q\}$;
Output: $\{Y_i \in \mathbb{O}^{n_i \times k}\}$, the set of most correlated orthonormal projection matrices.
1: $Y_i^{(0)} = []$ for $i = 1, 2, \ldots, \ell$;
2: call MATLAB's `svds` to compute the most dominant left singular vector $\boldsymbol{z} = [z_1^{\mathrm{T}}, \ldots, z_\ell^{\mathrm{T}}]^{\mathrm{T}}$ of $\mathscr{S}^{(0)}$, where $z_i \in \mathbb{R}^{n_i}$, and matrix-vector products by $\mathscr{S}^{(0)}$ and by $(\mathscr{S}^{(0)})^{\mathrm{T}}$ are calculated according to (27) and (28), respectively;
3: $y_i^{(1)} = z_i/\|z_i\|_2$ and $Y_i^{(1)} = y_i^{(1)}$ for $i = 1, 2, \ldots, \ell$;
4: **for** $j = 1, 2 \ldots, k-1$ **do**
5: call MATLAB's `svds` to compute the most dominant left singular vector $\boldsymbol{z} = [z_1^{\mathrm{T}}, \ldots, z_\ell^{\mathrm{T}}]^{\mathrm{T}}$ of $\mathscr{S}^{(j)}$, where $z_i \in \mathbb{R}^{n_i}$, and matrix-vector products by $\mathscr{S}^{(j)}$ and by $(\mathscr{S}^{(j)})^{\mathrm{T}}$ are calculated according to (27) and (28), respectively;
6: $y_i^{(j+1)} = z_i/\|z_i\|_2$ and $Y_i^{(j+1)} = [Y_i^{(j)}, y_i^{(j+1)}]$ for $i = 1, 2, \ldots, \ell$;
7: **end for**
8: $Y_i = Y_i^{(k)}$ for $i = 1, 2, \ldots, \ell$;
9: **return** $\{Y_i \in \mathbb{O}^{n_i \times k}\}$.

$$\mathscr{S}^{(j)} x, \quad (\mathscr{S}^{(j)})^{\mathrm{T}} y \tag{25}$$

fast, for any given $x \in \mathbb{R}^q$ and $y \in \mathbb{R}^n$, where $\mathscr{S}^{(j)}$ is given by (13) with S_i replaced by $S_i^{(j)}$ (introduced in the proof of Theorem 1). We now explain how these two matrix-vector products should be done for an efficient scalable implementation of Algorithm 1. Recall that S_i (denoted by $S_i^{(j)}$ hereafter) at line 4 there can be written as

$$S_i^{(j)} = \left[I_{n_i} - Y_i^{(j)} (Y_i^{(j)})^{\mathrm{T}} \right] S_i^{\mathrm{raw}} \left(I_q - \frac{1}{q} \mathbf{1}_q \mathbf{1}_q^{\mathrm{T}} \right), \tag{26}$$

where S_i^{raw} represents the original raw data matrix that may not even be centered, and $Y_i^{(j)}$ is given by (23). An efficient computation of $\mathscr{S}^{(j)} x =: z \equiv [z_1^{\mathrm{T}}, z_2^{\mathrm{T}}, \ldots, z_\ell^{\mathrm{T}}]^{\mathrm{T}}$ is as follows:

1) compute

$$x \leftarrow x - [(\mathbf{1}_q^{\mathrm{T}} x)/q] \mathbf{1}_q, \tag{27a}$$

2) for $i = 1, 2, \ldots, \ell$ do

$$z_i \leftarrow S_i^{\mathrm{raw}} x, \tag{27b}$$
$$z_i \leftarrow z_i - Y_i^{(j)} \left[(Y_i^{(j)})^{\mathrm{T}} x_i \right], \tag{27c}$$

where (27b) and (27c) are executed in order.

Likewise, $z := (\mathscr{S}^{(j)})^{\mathrm{T}} y$ should be done as follows: partition $y = [y_1^{\mathrm{T}}, y_2^{\mathrm{T}}, \ldots, y_\ell^{\mathrm{T}}]^{\mathrm{T}}$ with $y_i \in \mathbb{R}^{n_i}$, and then

1) for $i = 1, 2, \ldots, \ell$ do

$$y_i \leftarrow y_i - Y_i^{(j)} \left[(Y_i^{(j)})^{\mathrm{T}} y_i \right], \tag{28a}$$
$$y_i \leftarrow (S_i^{\mathrm{raw}})^{\mathrm{T}} y_i, \tag{28b}$$

where (28a) and (28b) are executed in order;

2) compute

$$y \leftarrow \sum_{i=1}^{\ell} y_i; \quad z \leftarrow y - [(\mathbf{1}_q^{\mathrm{T}} y)/q] \mathbf{1}_q. \tag{28c}$$

TABLE 1
Data sets used in the experiments, where the number of features for each view is shown inside the bracket.

Data set	samples training	samples testing	class	view 1	view 2	view 3	view 4	view 5	view 6
TVGraz	500	1558	10	Text (100)	Image (1024)	-	-	-	-
Wikipedia	693	2173	10	Text (100)	Image (1024)	-	-	-	-
Pascal	300	700	20	Text (100)	Image (1024)	-	-	-	-
Multiple-Features	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)	-
NUSWIDEOBJ	30000		31	CH (65)	CM (226)	CORR (145)	EDH (74)	WT (129)	-
Reuters	18758		6	English(21531)	France (24892)	German (34251)	Italian (15506)	Spanish (11547)	-

It is very important to notice that during (27) and (28), S_i^{raw} as the raw input matrices are never changed.

In our implementation of the incremental algorithm for UMvPLS for large scale (possibly sparse) multi-view data, we take advantage of MTALAB's `svds`, an implicit-restarted Lanczos type method based on the Golub-Kahan bidiagonlization. For more details about the mathematics and algorithm of `svds`, the reader is referred to [38], [39]. Algorithm 2 summarizes our current implementation for large-scale multi-view data.

Compared to the deflation method used in [30] for OMCCA-SS, Algorithms 1 and 2 have three major advantages:

1) Our algorithms are robust and efficient because it uses proven numerical linear algebra techniques as building blocks;
2) They do not require any C_{ii} to be positive definite because it relies on solving the standard symmetric eigenvalue problem like (19).
3) Algorithm 2 is scalable in handling large-scale high-dimensional multi-view data.

4 Experiments

We conduct experiments to compare the proposed UMvPLS (solved by our incremental algorithm) with baseline methods on various datasets in terms of two applications: cross-modal retrieval and multi-view feature extraction. As UMvPLS is devised for multi-view data, it is expected to work well for two view data, as well as data of more than two views. In the following, we will first evaluate UMvPLS for cross-modal retrieval due to its inherent problem of two views, and then conduct extensive experiments for multi-view feature extractions with various number of views.

4.1 Experimental settings

Datasets used in the experiments are shown in Table 1. The first three datasets [40]: TVGraz, Wikipedia, and Pascal, each of which consists of pairs of image and text, are used for the task of cross-modal retrieval. Specifically, there are 2058 pairs from 10 categories in TVGraz, 2866 pairs from 10 categories in Wikipedia, and 1000 pairs from 20 categories in Pascal. Images are represented by the bag-of-words (BOW) model using SIFT descriptors quantized with the 1024 visual word codebook [41] and texts are represented by the probabilities of text words under 100 hidden topics from latent Dirichlet allocation model [42]. As discussed in [40], the three datasets have different properties: Wikipedia contains high quality images and texts, but the image intra-class variability is large because of the broad class categories, so the classification accuracy is generally low on images; TVGraz has good correlations of image and text to the narrow object classes, so accuracies on both image and text are acceptable; Pascal is the most challenging visual datasets, where the added text features are not semantically rich, so accuracies on both image and text are low.

The rest six datasets in Table 1 are used for the task of multi-view feature extraction. We apply various feature descriptors to extract features of views for image datasets: Caltech101[1] [43], Scene15[2] [44], CENTRIST [45], GIST [46], LBP [47], histogram of oriented gradient (HOG), color histogram (CH), and SIFT-SPM [44]. Note that we drop CH for Scene15 due to the gray-level images. Multiple-Features is handwritten numeral data[3] [48] 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). NUSWIDEOBJ contains $30,000$ images from 31 categories, where five precomputed features are used: color moment (CM), CH, color correlation (CORR), edge distribution (EDH) and wavelet texture (WT) [49]. Reuters is a multi-view text categorization test collection dataset containing feature characteristics of documents originally written in five languages (English, French, German, Italian, and Spanish) and their translations over a common set of six categories (C15, CCAT, E21, ECAT, GCAT, and M11). Only a subset of Reuters, those written in English and their translations in other four languages, is used. The six datasets show different properties such as Retuters has high-dimensional features for each of five views and NUSWIDEOBJ consists of a large number of samples with a small number of related features for each view.

To demonstrate the effectiveness of Algorithm 2, we compare UMvPLS with existing methods CCA [6] and CCA-SSY [50] for the two-view datasets (the first three in Table 1), and MCCA with the SUMCOR model [29] and OMCCA-SS [30] for the multi-view dataset (the last six in Table 1). Except for the dimension k of the reduced space, all methods do not have any other hyper-parameter. In addition to the task-specific performance, we will also report the performance of the compared methods by varying the dimension k of the reduced space. All views of each dataset are mapped

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

TABLE 2
MAP scores of CCA, OCCA-SSY, and UMvPLS in terms of three metrics and two settings: image query and text query. The average MAP scores over the two settings are also reported. The best average scores over the three methods and three metrics are printed in bold, where the reduced dimension corresponding to the MAP score is shown in the bracket.

data	metric	CCA			OCCA-SSY			UMvPLS		
		Image Query	Text Query	Average	Image Query	Text Query	Average	Image Query	Text Query	Average
Wikipedia	L2	0.1255 (15)	0.1121 (10)	0.1188	0.1188 (1)	0.1553 (20)	0.1370	0.1499 (2)	0.1702 (4)	0.1601
	L1	0.1254 (20)	0.1121 (11)	0.1187	0.1188 (1)	0.1484 (20)	0.1336	0.1495 (2)	0.1614 (4)	0.1554
	NC	0.1163 (2)	0.1121 (5)	0.1142	0.1908 (20)	0.1573 (20)	0.1740	0.2445 (3)	0.1773 (6)	**0.2109**
TVGraz	L2	0.1132 (14)	0.1154 (3)	0.1143	0.1116 (1)	0.1116 (1)	0.1116	0.1772 (3)	0.2290 (9)	0.2031
	L1	0.1132 (16)	0.1157 (3)	0.1144	0.1116 (1)	0.1116 (1)	0.1116	0.1718 (3)	0.2135 (8)	0.1926
	NC	0.1123 (4)	0.1158 (1)	0.1141	0.1116 (1)	0.1116 (1)	0.1116	0.2764 (9)	0.2494 (9)	**0.2629**
Pascal	L2	0.0653 (3)	0.0584 (5)	0.0619	0.0863 (3)	0.1252 (3)	0.1058	0.1102 (3)	0.1208 (6)	0.1155
	L1	0.0645 (5)	0.0586 (20)	0.0616	0.0853 (3)	0.1205 (3)	0.1029	0.1064 (3)	0.1128 (5)	0.1096
	NC	0.0655 (20)	0.0599 (2)	0.0627	0.1576 (19)	0.1274 (19)	0.1425	0.1662 (4)	0.1320 (4)	**0.1491**

to the common space. Finally, the learning methods and evaluation approaches are conducted in the common space.

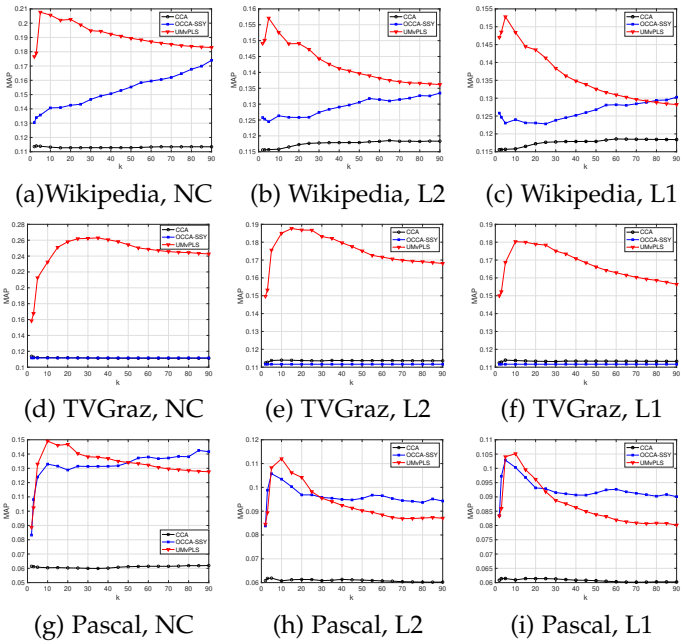
4.2 Cross-modal retrieval

Because of the rapid growth of multimedia data that usually contain mixtures of things, such as documents with both texts and images, any capability of decent cross-modal retrieval is significant and, as a result, cross-modal retrieval is attracting more and more attention. However, the inconsistency between different media types makes it challenging to measure the cross-media similarity of instances [51]. In overcoming the heterogeneity gap, CCA has become the standard approach by seeking a common representation for different media types [1], [2], [52].

Following [1], we consider two tasks given a set of pairs of text and image. One is the text retrieval using an image query, and the other is the image retrieval using a text query. For both tasks, the mean average precision (MAP) is used to measure the performance of the ranking produced by each CCA model. Note that MAP is a widely used measure in the image retrieval literature. The larger the MAP is, the better the model performs. Three distances, the L1 distance, the normalized correlation (NC) and the L2 distance, are evaluated by computing the similarity between a query and its retrieved object mapped into the common space produced by CCA and its variants.

Table 2 shows the best MAP and average scores for $k \in [2, 3, 5 : 5 : 90]$ by CCA, OCCA-SSY, and UMvPLS for the two tasks on the datasets Wikipedia, TVGraz, and Pascal. From Table 2, we have the following observations:

- Both OCCA-SSY and UMvPLS outperform CCA on Wikipedia and Pascal. UMvPLS is significantly more accurate than any of the two CCA-based methods for both tasks, especially on the most challenging data Pascal. This demonstrates that UMvPLSA is most effective for cross-modal retrieval.
- OCCA-SSY performs worst on TVGraz. Its MAP scores for both tasks are equally bad. One of the reason is that OCCA-SSY encountered numerical instability and, as a result, cannot obtain proper orthonormal projection matrices for both views. But UMvPLS does not face this issue. This empirically verifies the theoretical results in Section 3.3.
- Among all three methods with three distance metrics, UMvPLS shows the best results for both tasks.
- The MAP scores with NC for all three methods are the best among with any of the three metrics. This is consistent with the observations in [1]. Additionally, the ones by UMvPLS are the best among all three methods and that further proves that UMvPLS can maximize the correlation between image and text better.



(a)Wikipedia, NC (b) Wikipedia, L2 (c) Wikipedia, L1

(d) TVGraz, NC (e) TVGraz, L2 (f) TVGraz, L1

(g) Pascal, NC (h) Pascal, L2 (i) Pascal, L1

Fig. 1. Average MAP scores of CCA, OCCA-SSY, and UMvPLS over two tasks on datasets: Wikipedia, TVGraz, and Pascal with respect to NC, L2, and L1.

We also demonstrate the robustness of the three methods as the dimension k of the reduced space varies. Fig. 1 illustrates the average MAP scores of the compared methods over two tasks on the three datasets with respect to metrics NC, L2, and L1. In all cases, CCA performs very poorly and is a noncontender, compared to the better one of UMvPLS and OCCA-SSY. On TVGraz, UMvPLS outperforms OCCA-SSY by wide margins; On Wikipedia, UMvPLS achieves significantly better results than OCCA-SSY for smaller tested ks but their performance gap starts to shrink as k increases; On Pascal, UMvPLS holds an edge over OCCA-SSY for small ks ($k \leq 50$ for NC, $k \leq 30$ for L2, and $k \leq 25$ for L1). The peak average MAP scores of UMvPLS can generally be obtained for $k \in [2, 30]$, but OCCA-SSY often requires larger k for decent average MAP scores. Hence, UMvPLS is more suitable than OCCA-SSY for retrieval and achieves faster responses since computing similarity between two points in

TABLE 3
Classification accuracy of MCCA, OMCCA-SS, UMvPLS, and the single-view classifier with 20% training and 80% testing split on data.

	Multiple-features	Caltech101-7	Caltech101-20	NUSWIDEOBJ	Scene15	Reuters
view1	0.9434 ± 0.0056	0.9214 ± 0.0049	0.7543 ± 0.0056	0.1597 ± 0.0023	0.5537 ± 0.0108	0.5019 ± 0.0312
view2	0.7396 ± 0.0105	0.9312 ± 0.0064	0.8023 ± 0.0056	0.1733 ± 0.0033	0.5100 ± 0.0099	0.4784 ± 0.0271
view3	0.9133 ± 0.0081	0.9359 ± 0.0065	0.8017 ± 0.0066	0.1915 ± 0.0019	0.5385 ± 0.0051	0.4537 ± 0.0453
view4	0.6731 ± 0.0124	0.9080 ± 0.0126	0.7760 ± 0.0126	0.1698 ± 0.0022	0.4478 ± 0.0072	0.4556 ± 0.0318
view5	0.9530 ± 0.0059	0.7504 ± 0.0082	0.5816 ± 0.0118	0.2011 ± 0.0013	0.6484 ± 0.0174	0.4503 ± 0.0282
view6	0.7731 ± 0.0110	0.9111 ± 0.0061	0.7404 ± 0.0121	-	-	-
MCCA	0.8666 ± 0.0064(6)	0.8750 ± 0.0099(15)	0.8427 ± 0.0070(40)	0.2130 ± 0.0037(30)	0.6696 ± 0.0095(30)	0.7616 ± 0.0040(25)
OMCCA-SS	0.8198 ± 0.0077(6)	0.9390 ± 0.0083(45)	0.8329 ± 0.0057(50)	0.1190 ± 0.0000(3)	0.6792 ± 0.0085(50)	0.7825 ± 0.0026(45)
UMvPLS	$\mathbf{0.9599 \pm 0.0037}$(5)	$\mathbf{0.9525 \pm 0.0049}$(10)	$\mathbf{0.8621 \pm 0.0079}$(20)	$\mathbf{0.2737 \pm 0.0025}$(10)	$\mathbf{0.6801 \pm 0.0067}$(15)	$\mathbf{0.7949 \pm 0.0028}$(30)

reduced spaces of lower-dimensions is much cheaper.

4.3 Multi-view feature extraction

We evaluate the performance of UMvPLS and two other CCA-based multi-view learning methods, MCCA and OMCCA for the multi-view feature extraction task in terms of classification. By following the experimental settings in [30], we divide each experiment in four steps: 1) split data into training and testing data with certain ratio, 2) learn the mapping function from the training data to map the data of each view to a reduced common space, 3) obtain the representations of multi-view data points for both training and testing data via the serial feature fusion strategy [19], and 4) predict/evaluate the labels of testing data via the 1-nearest neighbor (1NN) classifier learned from the training data. In the experiments, we use the Euclidean distance as the distance metric in the 1NN classifier and classification accuracy to measure learning performance. As in our cross-modal retrieval experiments, we also investigate the impact of the dimension k of the reduced space. Moreover, we study the influence of the ratio of training and testing data split. To achieve statistically meaningful results, we repeat each experiment with 10 random splits of training and testing data and report the mean accuracies with the associated standard deviations. In addition, we evaluate the performance of the 1NN classifier using its results on each single view as the baselines. It is worth noting that Reuters contains large-scale high-dimensional data. MCCA and OMCCA-SS cannot handle such high-dimensional data due to the high computational complexity of eigen-decomposition and large memory requirement. To make them feasible and use their results as the baselines for Reuters, we apply PCA to each of its view and to reduce their dimensions to 1000. Since UMvPLS is efficient for and can tackle data of very high dimensions (even more so when all S_i^{raw} are sparse), we don't need and don't do such a preprocessing step when it comes to UMvPLS.

The classification accuracies of MCCA, OMCCA-SS, and UMvPLS on six multi-view datasets are shown in Table 3, together with the classification results obtained by each of the single-view data, respectively. From Table 3, we have the following observations:

- The classification results based on each single view data can be very different viewwise. MCCA achieves better accuracies on four out of the six datasets than the best single-view classifier results. On the other two datasets, Multiple-features and Caltech101-7, MCCA is still better than the worst single-classifier.
- OMCCA-SS also achieves better accuracies on four out of the six datasets than the best single-view classifier, but not all of the four datasets on which OMCCA-SS performs better are the same as the four datasets on which MCCA performs better. Overall, OMCCA-SS and MCCA are comparable and, in fact, each method beats the other on three out of the six datasets. It is also clear to see that OMCCA-SS can achieve much better result than MCCA on Caltech101-7, but it performs worse on Multiple-features.
- UMvPLS obtains the best accuracies over all six datasets. It not only demonstrates the ability to improve the best single-view classifier, but also shows significantly better results than MCCA and OMCCA-SS. Another interesting phenomenon is that the peak performance of UMvPLS as k varies is often achieved at a relatively small k. That leads to an important practical consequence, i.e., to use UMvPLS with a relatively small k for the best performance and yet at the lowest cost.

These observations demonstrate that UMvPLS improves classification performance for multi-view feature extraction over MCCA, OMCCA-SS, and the single-view classifier. Compared to OMCCA-SS, UMvPLS can obtain better accuracies possibly for two reasons: 1) UMvPLS can better maximize variance matrices, and 2) OMCCA-SS has a numerical instability issue and indeed it fails on Multiple-features. UMvPLS is built on proven numerical linear algebra techniques and guarantees to produce numerically orthonormal projection matrices (Theorem 1).

We further investigate the influence of three important factors to the multi-view feature extraction, including:

1) how does the reduced dimension k affect the classification performance?
2) how does the size of training data relative to whole data affect the classification performance?
3) how does the reduced dimension k affect the CPU time?

Fig. 2 displays our numerical results with respect to these three factors on five of the six datasets.

First, we observe that accuracy increases when the ratio of training data increases, as one might expect. On some datasets such as Caltech101-7 and Caltech101-20 (view 5), the accuracies obtained by the single-view classifier from some of the views are noticeably lower than from other views. These views may be considered as weak views for the purpose. The improvements of MCCA and OMCCA-SS over the single-view classifier are quite dramatic on Reuters, for which the accuracies by the single-view classifier on each view are all much poorer. UMvPLS produces the best results over all datasets and the tested training ratios, except for the single-view classifier on view 5 for Scene15 with training

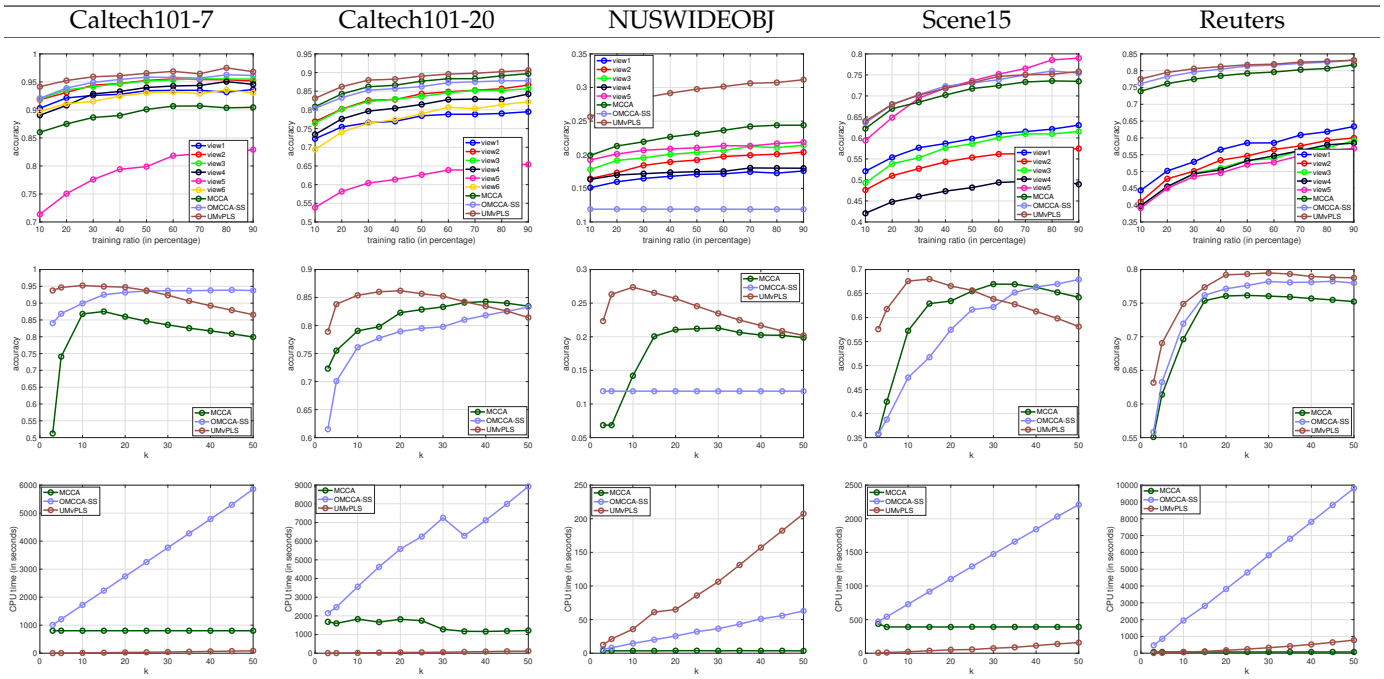


Fig. 2. Accuracies with respect to the ratio of training data over all tested ks (the first row), the dimension k with 20% training data (the second row), as well as the empirical computational cost in CPU time on five multi-view datasets (the third row).

ratio larger than 60%. These results confirm that multi-view feature extraction using subspace learning can work better, and often much better, than the best single-view classifier.

Second, the accuracies of MCCA, OMCCA-SS, UMvPLS behave very differently from one another as k varies. UMvPLS often obtains the peak accuracy at a relatively small k, while CCA and OMCCA-SS need a large reduced dimension k in order to achieve reasonable accuracies that are still worse than UMvPLS at a much smaller k. In particular, OMCCA-SS fails on NUSWIDEOBJ.

Third, UMvPLS takes much less CPU time on high-dimensional data, except for NUSWIDEOBJ for a good reason to be explained in a moment. As discussed in Section 3.4, UMvPLS has a computational complexity that is linear in the number of input samples, so the reported CPU times are consistent with the observed computational times on NUSWIDEOBJ. On the other hand, UMvPLS is also linear in the input dimension of the sample points, so it is much cheaper than $O(n^3)$, the computational complexity of MCCA, and $O(kn^3)$, the computational complexity of OMCCA-SS, where $n = \sum_{i=1}^{\ell} n_i$. As observed, UMvPLS runs fastest on the rest of the four datasets.

We now explain why UMvPLS uses more time on NUSWIDEOBJ than the other two methods. In fact, for NUSWIDEOBJ, the total number of features $n = \sum_i n_i$ in all views is 639 while the number q of samples in each view is $30,000$ which is much bigger than 639. In such a case, the more efficient way is to form covariance matrix $\mathscr{A} = \mathscr{S}\mathscr{S}^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ explicitly once and for all at a cost $O(n^2q)$ and then work with $\mathscr{A}$ exclusively afterwards at a cost $O(n^2)$. This idea helps only when n is modest and $n \ll q$. But for n so large that storing $n \times n$ dense matrices becomes an issue, the idea will not work even if $n \ll q$. In this latter case, UMvPLS (solved by Algorithm 2) can still get the job done if all S_i^{raw} are sparse.

In summary, the proposed method UMvPLS not only outperforms baseline methods, but also runs faster on and are feasible for high-dimensional data. It is scalable for unsupervised subspace learning on large-scale high-dimensional multi-view data because of its linear computational complexity in the number of nonzero entries in the given dataset.

5 Conclusion

In this paper, we study an extension of the partial least squares (PLS) method for multi-view data in the unsupervised setting. A new method called the unsupervised multi-view partial least squares (UMvPLS) method is presented and it is inspired by the analogous study to CCA and its variants. To solve the resulting challenging optimization problem, we propose a stable deflation approach with theoretical guarantee and further adapt it to efficiently handle large-scale high-dimensional data, based on well-developed matrix computational techniques. Our experimental results for two learning tasks, cross-modal retrieval and multi-view feature extraction, show that our new method outperforms existing methods and is scalable for multi-view data.

References

[1] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proceedings of the 18th ACM International Conference on Multimedia*. ACM, 2010, pp. 251–260.

[2] Y. Peng, X. Huang, and Y. Zhao, "An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2372–2385, 2017.

[3] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[4] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[5] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, vol. 38, pp. 43–54, 2017.

[6] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3-4, pp. 321–377, 1936.

[7] S. Wold, C. Albano, W. Dunn, U. Edlund, K. Esbensen, P. Geladi, S. Hellberg, E. Johansson, W. Lindberg, and M. Sjöström, "Multivariate data analysis in chemistry," in *Chemometrics*. Springer, 1984, pp. 17–95.

[8] W. Yang, Y. Shi, Y. Gao, L. Wang, and M. Yang, "Incomplete-data oriented multiview dimension reduction via sparse low-rank representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 6276–6291, 2018.

[9] L. Xie, W. Guo, H. Wei, Y. Tang, and D. Tao, "Efficient unsupervised dimension reduction for streaming multiview data," *IEEE Transactions on Cybernetics*, 2020.

[10] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[11] G. Cao, A. Iosifidis, K. Chen, and M. Gabbouj, "Generalized multi-view embedding for visual recognition and cross-modal retrieval," *IEEE Transactions on Cybernetics*, vol. 48, no. 9, pp. 2542–2555, 2017.

[12] M. Xu, Z. Zhu, X. Zhang, Y. Zhao, and X. Li, "Canonical correlation analysis with l2, 1-norm for multiview data representation," *IEEE Transactions on Cybernetics*, 2019.

[13] A. Benton, H. Khayrallah, B. Gujral, D. A. Reisinger, S. Zhang, and R. Arora, "Deep generalized canonical correlation analysis," *arXiv preprint arXiv:1702.02519*, 2017.

[14] V. Uurtio, J. M. Monteiro, J. Kandola, J. Shawe-Taylor, D. Fernandez-Reyes, and J. Rousu, "A tutorial on canonical correlation methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–33, 2017.

[15] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 483–502, 2005.

[16] E. Kokiopoulou and Y. Saad, "Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2143–2156, 2007.

[17] D. Cai and 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.

[18] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. Mach. Learning Res.*, vol. 16, pp. 2859–2900, 2015.

[19] X.-B. Shen, Q.-S. Sun, and Y.-H. Yuan, "Orthogonal canonical correlation analysis and its application in feature fusion," in *Proceedings of the 16th International Conference on Information Fusion*. IEEE, 2013, pp. 151–157.

[20] L. Wang, L.-h. Zhang, Z. Bai, and R.-C. Li, "Orthogonal canonical correlation analysis and applications," *Optimization Methods and Software*, vol. 35, no. 4, pp. 787–807, 2020.

[21] L. Zhang, L. Wang, Z. Bai, and R. cang Li, "A self-consistent-field iteration for orthogonal cca," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[22] J. Arenas-Garcıa and K. B. Petersen, "Kernel multivariate analysis in remote sensing feature extraction," *Kernel Methods for Remote Sensing Data Analysis*, p. 329, 2009.

[23] A. Sharma and D. W. Jacobs, "Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch," in *CVPR 2011*. IEEE, 2011, pp. 593–600.

[24] Y. Mou, L. Zhou, X. You, Y. Lu, W. Chen, and X. Zhao, "Multi-view partial least squares," *Chemometrics and Intelligent Laboratory Systems*, vol. 160, pp. 13–21, 2017.

[25] P. Horst, "Generalized canonical correlations and their applications to experimental data," *J. Clinical Psychology*, vol. 17, no. 4, pp. 331–347, 1961.

[26] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[27] J. Yin and S. Sun, "Multiview uncorrelated locality preserving projection," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[28] W. Liu, X. Yang, D. Tao, J. Cheng, and Y. Tang, "Multiview dimension reduction via hessian multiset canonical correlations," *Information Fusion*, vol. 41, pp. 119–128, 2018.

[29] A. A. Nielsen, "Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 293–305, 2002.

[30] X. Shen and Q. Sun, "Orthogonal multiset canonical correlation analysis based on fractional-order and its application in multiple feature extraction and recognition," *Neural Processing Letters*, vol. 42, no. 2, pp. 301–316, 2015.

[31] D. Chu, L. Liao, M. K. Ng, and X. Zhang, "Sparse kernel canonical correlation analysis," in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2013*, ser. IMECS 2013, vol. I, Hong Kong, March 2013.

[32] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge: Cambridge University Press, 1991.

[33] H. Wielandt, "Das Iterationsverfahren bei nicht selbstadjungierten linearen Eigenwertaufgaben," *Math. Z.*, vol. 50, no. 1, pp. 93–143, 1944.

[34] J. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA: SIAM, 1997.

[35] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, Maryland: Johns Hopkins University Press, 2013.

[36] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed. Philadelphia: SIAM, 1999.

[37] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (editors), *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide*. Philadelphia: SIAM, 2000.

[38] R. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User's Guide*. Philadelphia: SIAM, 1998.

[39] D. C. Sorensen, "Implicit application of polynomial filters in a k-step Arnoldi method," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 1, pp. 357–385, 1992.

[40] J. C. Pereira and N. Vasconcelos, "On the regularization of image semantics by modal expansion," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3093–3099.

[41] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[42] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[43] F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[44] S. Lazebnik, C. Schmid, and 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.

[45] J. Wu and 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.

[46] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[47] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 971–987, 2002.

[48] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[49] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: a real-world web image database from National University of Singapore," in *Proceedings of the ACM International Conference on Image and Video Retrieval*. ACM, 2009, p. 48.

[50] X. Shen, Q. Sun, and 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.

[51] J. Chi, X. Huang, and Y. Peng, "Zero-shot cross-media retrieval with external knowledge," in *International Conference on Internet Multimedia Computing and Service*. Springer, 2017, pp. 200–211.

[52] F. Wu, H. Zhang, and Y. Zhuang, "Learning semantic correlations for cross-media retrieval," in *2006 International Conference on Image Processing*. IEEE, 2006, pp. 1465–1468.

Li Wang is currently an assistant professor with Department of Mathematics and Department of Computer Science and Engineering, University of Texas at Arlington. She received her Ph.D. degree in Mathematics from University of California at San Diego in 2014, her master degree in Computational Mathematics from Xi'an Jiaotong University (Shaanxi, China) in 2009, and her Bachelor degree in Information and Computing Science from China University of Mining and Technology (Jiangsu, China) in 2006. She worked as a research assistant professor with Department of Mathematics, Statistics, and Computer Science at University of Illinois at Chicago from 2015 to 2017, a postdoctoral fellow at University of Victoria (British Columbia, Canada) in 2015, and at Brown University in 2014. Her research interests include large scale optimization, polynomial optimization and machine learning.

Ren-Cang Li is currently a professor with the Department of Mathematics, University of Texas at Arlington. He received his BS degree in computational mathematics from Xiamen University (Fujian, China) in 1985, his MS degree, also in computational mathematics, from the Chinese Academy of Science (Beijing, China) in 1988, and his PhD degree in applied mathematics from University of California at 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 a CAREER award from the US National Science Foundation 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.