A Self-Consistent-Field Iteration for Orthogonal Canonical Correlation Analysis

Lei-Hong Zhang, Li Wang, Zhaojun Bai, and Ren-Cang Li

Abstract—We propose an efficient algorithm for solving orthogonal canonical correlation analysis (OCCA) in the form of trace-fractional structure and orthogonal linear projections. Even though orthogonality has been widely used and proved to be a useful criterion for visualization, pattern recognition and feature extraction, existing methods for solving OCCA problem are either numerically unstable by relying on a deflation scheme, or less efficient by directly using generic optimization methods. In this paper, we propose an alternating numerical scheme whose core is the sub-maximization problem in the trace-fractional form with an orthogonality constraint. A customized self-consistent-field (SCF) iteration for this sub-maximization problem is devised. It is proved that the SCF iteration is globally convergent to a KKT point and that the alternating numerical scheme always converges. We further formulate a new trace-fractional maximization problem for orthogonal multiset CCA and propose an efficient algorithm with an either Jacobi-style or Gauss-Seidel-style updating scheme based on the SCF iteration. Extensive experiments are conducted to evaluate the proposed algorithms against existing methods, including real-world applications of multi-label classification and multi-view feature extraction. Experimental results show that our methods not only perform competitively to or better than the existing methods but also are more efficient.

Index Terms—canonical correlation analysis; self-consistent-field iteration; orthogonal multiset canonical correlation analysis.

1 Introduction

Canonical correlation analysis (CCA) [1], [2] is a standard statistical technique and widely-used feature extraction paradigm for two sets of multidimensional variables. It finds basis vectors for the two sets of variables such that the correlations between the projections of the variables onto these basis vectors are mutually maximized. During the last decade, CCA has received a renewed interest in the machine learning community and its applicability has been demonstrated in various fields [3].

In this paper, we are particularly interested in a variant of CCA, namely orthogonal CCA (OCCA), in which projections are constrained to be orthogonal [2], [4]. Distinguished from the classical CCA, OCCA has its exclusive property of preserving the covariance of the original data [2]. In addition, OCCA inherits many advantages such as being less sensitive to noise, better suited for data visualization and preserving metrics, brought by various other learning models for pattern recognition and feature extraction, where the orthogonality has been proved as an effective learning criterion. For examples, orthogonal linear discriminant analysis (LDA) is observed to have better performance than the standard LDA since the orthogonality constraint to some extent can remove noise [5]; orthogonal neighborhood preserving projections [6] achieves better representation of the global structure and is effective for data visualization; orthogonal locality preserving indexing [7] shares the same locality preserving character as locality preserving indexing and at the same time requires the basis functions to be orthogonal so that the metric structure of the document space is preserved. OCCA has further been extended for more than two views [8].

Comparing to CCA, OCCA brings the above-mentioned advantages for data analysis, but it no longer retains an analytic solution. A common heuristic approach is to orthogonalize the basis vectors obtained by CCA. However, this produces a suboptimal solution for the OCCA problem. In [4], an incremental scheme is employed to produce current basis vectors with additional constraints to enforce the orthogonality with the previously computed basis vectors. We will point out in Section 2 that the incremental scheme relies on a generalized eigenvalue problem that is numerically unstable in the sense that the theoretically dominant eigenvalue, although provably real, could be numerically computed to be complex and as a result, no real basis vector can be found and the scheme breaks down. When that happens, some kind of post-processing step is required to obtain a feasible solution [4]. In [2], generic optimization methods for minimizing a smooth function over the product of the Stiefel manifolds are used for OCCA. These methods usually converge to a local minimizer but they do not take the trace-fractional structure of OCCA into consideration. As a result, they are usually less efficient than custom-made algorithms. These challenges hinder OCCA from being widely used.

- *Lei-hong Zhang is with School of Mathematical Sciences and Institute of Computational Science, Soochow University, Suzhou 215006, Jiangsu, China. Supported in part by National Natural Science Foundation of China NSFC-11671246, National Key R&D Program of China (No. 2018YFB0204404) and 2018 Double Innovation Program of Jiangsu Province, China. Email: longzlh@suda.edu.cn.*
- *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.*
- *Zhaojun Bai is with Department of Computer Science and Department of Mathematics, University of California, Davis, CA 95616, USA. Supported in part by NSF DMS-1913364. Email: zbai@ucdavis.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.*

The goal of this paper is to propose new efficient algorithms for solving the OCCA problem with guaranteed theoretical convergence and numerical stabilizability. In order to fully explore the trace-fractional structure of OCCA, we first uncover the connection of OCCA with an eigenvector-dependent nonlinear eigenvalue problem (NEPv), and then naturally come up with a simple iterative method whose numerical efficiency is guaranteed by a structure-exploiting self-consistent-field (SCF) iteration. Global convergence and local convergence of this customized algorithm are established.

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

- We propose a novel algorithm OCCA-scf for solving OCCA in the form of a trace-fractional matrix optimization problem. The proposed algorithm is built upon an efficient and effective SCF iteration to solve a very special trace-ratio sub-maximization problem through taking the trace-fractional structure into account. It is proved that the SCF iteration is always convergent and, as a result, OCCA-scf is guaranteed to converge. It can also integrate the state-of-the-art eigensolvers within the iteration framework for large scale problems. Moreover, it guarantees the orthogonality of the computed basis vectors.
- We present a new orthogonal multiset CCA (OMCCA) model with integrated weights for each pair of views and the trace-fractional objective for correlations between any two views. By leveraging the same customized SCF iteration, a novel range constrained OMCCA algorithm is proposed with an either Jacobi-style or Gauss-Seidel-style updating scheme.
- Extensive experiments are conducted for evaluating the proposed algorithms against existing methods in terms of various measurements, including sensitivity analysis, correlation analysis, computation analysis, and data visualization. We further apply our methods for two real world applications: multi-label classification and multi-view feature extraction. Experimental results show that our methods not only perform competitively to or better than baselines but also are more efficient.

Paper organization. We first review classical CCA models and existing OCCA variants in Section 2. In Section 3, we propose a novel algorithm for solving OCCA problem and the main theoretical results are presented in Section 4. In Section 5, we develop a new algorithm for OMCCA by leveraging the same SCF iteration. Extensive experiments are conducted in Section 6. Finally, we draw our conclusions in Section 7. All proofs are given in the supplementary material.

Notation. $\mathbb{R}^{m\times n}$ is the set of $m\times n$ real matrices and $\mathbb{R}^n = \mathbb{R}^{n\times 1}$. $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix, and $\mathbf{1}_n \in \mathbb{R}^n$ is the vector of all ones. $\|\mathbf{x}\|_2$ is the 2-norm of a vector $\mathbf{x}\in\mathbb{R}^n$. For $B\in\mathbb{R}^{m\times n}$, $\mathcal{R}(B)$ is the column subspace and its singular values are denoted by $\sigma_i(B)$ for $i = 1,\ldots,\min(m,n)$ arranged in the nonincreasing order. $\|B\|_1$ ($\|B\|_2$) is the 1-norm (2-norm) of matrix B. The thin SVD of B is the one $B = U\Sigma V^{\mathrm{T}}$ such that $\Sigma = \mathrm{Diag}(\sigma_1(B),\ldots,\sigma_r(B))$ is diagonal with $r = \mathrm{rank}(B)$, the rank of B, and $\|B\|_{\mathrm{tr}} = \sum_{i=1}^r \sigma_i(B)$ is its trace norm (also known as the nuclear norm). When $m = n$, $\mathrm{sym}(B) = (B + B^{\mathrm{T}})/2$; if B is also symmetric, then $\mathrm{eig}(B) = \{\lambda_i(B)\}_{i=1}^n$ denotes the set of its eigenvalues (counted by multiplicities) arranged in the ~~nonincreasing~~ order, and $B \succ 0 (\succeq 0)$ means that B is symmetric positive definite (semi-definite). The Stiefel manifold

$$\mathbb{O}^{n\times k} = \{X\in\mathbb{R}^{n\times k} \,:\, X^{\mathrm{T}}X = I_k\}$$

is an embedded submanifold of $\mathbb{R}^{n\times k}$ endowed with the standard inner product $\langle X, Y\rangle = \mathrm{tr}(X^{\mathrm{T}}Y)$ for $X, Y \in \mathbb{R}^{n\times k}$, where $\mathrm{tr}(X^{\mathrm{T}}Y)$ is the trace of $X^{\mathrm{T}}Y$. The tangent space $\mathcal{T}_X\mathbb{O}^{n\times k}$ of $\mathbb{O}^{n\times k}$ at $X\in\mathbb{O}^{n\times k}$ is given by (see, e.g., [9])

$$\mathcal{T}_X\mathbb{O}^{n\times k} = \{H\in\mathbb{R}^{n\times k} | X^{\mathrm{T}}H + H^{\mathrm{T}}X = 0\} \tag{1a}$$

$$= \left\{H\in\mathbb{R}^{n\times k}\middle| \begin{array}{l} H = XK + (I_n - XX^{\mathrm{T}})J \\ \forall K = -K^{\mathrm{T}}\in\mathbb{R}^{k\times k},\ J\in\mathbb{R}^{n\times k}\end{array}\right\}. \tag{1b}$$

2 Related Work

We review the classical CCA and OCCA methods, as well as their extensions to multiple sets of variables.

2.1 Classical CCA via SVD

CCA is a two-view multivariate statistical method [1], where the variables of observations can be partitioned into two sets, i.e., the two views of the data. Denote the data matrices $S_1\in\mathbb{R}^{n\times q}$ and $S_2\in\mathbb{R}^{m\times q}$ from view 1 and view 2 with n and m features, respectively, where q is the number of samples. Assume both S_1 and S_2 are centralized, i.e., $S_1\mathbf{1}_q = 0$ and $S_2\mathbf{1}_q = 0$; otherwise, we may preprocess S_i as $S_i \leftarrow S_i - (S_i\mathbf{1}_q)\mathbf{1}_q^{\mathrm{T}}/q$ for $i = 1, 2$. Let $\mathbf{x}_1\in\mathbb{R}^n$ and $\mathbf{x}_2\in\mathbb{R}^m$ be the canonical weight vectors. The canonical variates are the linear transformations defined as $\mathbf{z}_1 = S_1^{\mathrm{T}}\mathbf{x}_1, \mathbf{z}_2 = S_2^{\mathrm{T}}\mathbf{x}_2$. The canonical correlation between the two canonical variates is defined as $\rho(\mathbf{x}_1,\mathbf{x}_2) = \mathbf{z}_1^{\mathrm{T}}\mathbf{z}_2/(||\mathbf{z}_1||_2||\mathbf{z}_2||_2)$. CCA aims to find the pair of canonical weight vectors that maximize the canonical correlation:

$$\max_{\mathbf{x}_1,\mathbf{x}_2} \rho(\mathbf{x}_1,\mathbf{x}_2). \tag{2}$$

It can also be interpreted as the problem of finding the best pair of canonical weight vectors so that the cosine of the angle between the two canonical variates is maximized, that is, the smallest angle in $[0, \frac{\pi}{2}]$.

This single-vector CCA (2) has been extended to obtain the pair of canonical weight matrices, namely, the pair of canonical weight matrices $X_1\in\mathbb{R}^{n\times k}$ and $X_2\in\mathbb{R}^{m\times k}$ by solving the following optimization problem

$$\max_{X_1,X_2} \mathrm{tr}(X_1^{\mathrm{T}}C_{1,2}X_2),\ \text{s.t. } X_i^{\mathrm{T}}C_{i,i}X_i = I_k,\ i = 1, 2, \tag{3}$$

where $C_{i,j} = S_iS_j^{\mathrm{T}}$, $i, j = 1, 2$. Hereafter, (3) is referred to as the classical CCA, or simply CCA for short. In general, the closed-form solution of (3) can be obtained by the singular value decomposition (SVD) [10], and it can be proved that $X_1^{\mathrm{T}}C_{1,2}X_2 \succeq 0$ for any optimal solution pair (X_1, X_2) [11, Theorem III.2].

2.2 OCCA via generic methods over matrix manifolds

The classical CCA is not suitable for settings where an orthogonal projection is required in an orthogonal coordinate system, such as for data visualization. This is because optimal X_1 and X_2 in (3) usually do not have orthonormal columns. Although one can always orthogonalize their columns as a post-processing step, the resulting orthogonal projections are generally suboptimal. For that reason, an orthogonal CCA (OCCA) is proposed in [2], [4] to maximize the correlation

$$f(X_1, X_2) = \frac{\operatorname{tr}(X_1^{\mathrm{T}} C_{1,2} X_2)}{\sqrt{\operatorname{tr}(X_1^{\mathrm{T}} C_{1,1} X_1)\operatorname{tr}(X_2^{\mathrm{T}} C_{2,2} X_2)}} \tag{4}$$

directly over orthonormal matrices, i.e.,

$$\max_{X_1 \in \mathbb{O}^{n\times k}, X_2 \in \mathbb{O}^{m\times k}} f(X_1, X_2). \tag{5}$$

As pointed out in [2], OCCA is different from the classical CCA because OCCA preserves the covariance of the original data S_1 and S_2 by finding orthonormal matrices that maximize the correlation, while the classical CCA whitens each dataset and projects them so that the correlation is maximized.

Generic optimization methods for minimizing or maximizing a smooth function over the product of the Stiefel manifolds are available. Classical optimization algorithms such as the steepest descent gradient or the trust-region methods over the Euclidean space have been extended to the general Riemannian manifolds in [9]. However, besides only guaranteeing to converge to a local optimizer at best, these generic algorithms do not make use of the special trace-fractional structure in (5), and therefore, they usually are less efficient than custom-made algorithms for trace-ratio-related optimizations (see [12], [13] for numerical results of trace-ratio optimizations).

2.3 OCCA via a greedy method

The motivation of imposing orthogonality constraints was also explored in [4]. A greedy method (which we will call OCCA-SSY for short) is employed to find k pairs of orthogonal vectors, computed one pair at a time. The initial step is the same as the classical CCA to find the pair of canonical weight vectors $(\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)})$ that solves (2). Given $\{(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})\}_{t=1}^{r}$, the $(r+1)$st step is to solve the following problem

$$\begin{aligned}&(\mathbf{x}_1^{(r+1)}, \mathbf{x}_2^{(r+1)}) = \arg\max_{\mathbf{x}_1, \mathbf{x}_2} \ \rho(\mathbf{x}_1, \mathbf{x}_2)\\ &\text{s.t.} \quad \mathbf{x}_i^{\mathrm{T}} C_{i,i} \mathbf{x}_i = 1, \ \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_i^{(t)} = 0, \ i = 1, 2, \ t = 1, \ldots, r.\end{aligned}$$

Such an approach relies on a deflation scheme, and the pair $(\mathbf{x}_1^{(r+1)}, \mathbf{x}_2^{(r+1)})$ is claimed to correspond to the dominant eigenpair of a generalized eigenvalue problem, which, however, numerically may not have any real eigenpair and thus is unusable[1].

1. Private communications with the authors of [4], 2019.

2.4 Classical multiset CCA

Multiset CCA (MCCA) [14], [15] is proposed to analyze linear relationships among more than two canonical variates. It is a generalization of the classical CCA [1]. Here, we briefly introduce a widely used model [15] by seeking projections to maximize the sum of the pairwise correlations between any two canonical variates. Specifically, given ℓ datasets in the form of matrices

$$S_i \in \mathbb{R}^{n_i \times q} \quad \text{for } i = 1, 2, \ldots, \ell, \tag{6}$$

where n_i is the number of features in the ith dataset, and q is the number of data points in each of the datasets. Without loss of generality, we may assume that all S_i are centered, i.e., $S_i \mathbf{1}_q = 0$ for all i. Let $C_{i,j} = S_i S_j^{\mathrm{T}}$ for $i, j = 1, \ldots, \ell$. MCCA seeks to find the set of ℓ canonical weight vectors that solves

$$\max_{\mathbf{x}_1, \ldots, \mathbf{x}_\ell} \sum_{i,\, j=1}^{\ell} \mathbf{x}_i^{\mathrm{T}} C_{i,j} \mathbf{x}_j, \tag{7}$$

subject to

$$\text{either} \quad \sum_{i=1}^{\ell} \mathbf{x}_i^{\mathrm{T}} C_{i,i} \mathbf{x}_i = 1, \tag{8}$$

$$\text{or} \quad \mathbf{x}_i^{\mathrm{T}} C_{i,i} \mathbf{x}_i = 1, \ i = 1, \ldots, \ell. \tag{9}$$

KKT conditions for MCCA under either (8) or (9) can be found in [15]. In particular, under (8), MCCA (7) is equivalent to a generalized eigenvalue problem [15, p.297], which can be solved by an eigensolver [10], [16], [17].

2.5 OMCCA via a greedy method

In [8], a greedy orthogonal MCCA (OMCCA) (called OMCCA-SS for short) was proposed. Similar to [4], it goes as follows. Given $\{\mathbf{x}_i^{(t)}, \forall i = 1, \ldots, \ell\}$ for $t = 1, \ldots, r$, OMCCA-SS recursively solves the following subproblems

$$\begin{aligned}&\{\mathbf{x}_1^{(r+1)}, \ldots, \mathbf{x}_\ell^{(r+1)}\} = \arg\max_{\mathbf{x}_1, \ldots, \mathbf{x}_\ell} \ \sum_{i,\, j=1}^{\ell} \mathbf{x}_i^{\mathrm{T}} C_{i,j} \mathbf{x}_j,\\ &\text{s.t.} \quad \text{(8) and } \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_i^{(t)} = 0, \ i = 1, \ldots \ell, t = 1, \ldots, r.\end{aligned}$$

OMCCA-SS inherits the same issues as OCCA-SSY discussed in Subsection 2.3.

3 Novel Algorithm for OCCA

In this section we propose a new optimization scheme for solving the OCCA problem (5) by fully taking the advantage of its underlying structure.

3.1 Reformulation of OCCA

Following the annotation in Section 2, both views of the data S_1 and S_2 are centralized in advance. Define

$$A = S_1 S_1^{\mathrm{T}} \in \mathbb{R}^{n\times n}, \ B = S_2 S_2^{\mathrm{T}} \in \mathbb{R}^{m\times m}, \ C = S_1 S_2^{\mathrm{T}} \in \mathbb{R}^{n\times m}.$$

Let $X \in \mathbb{O}^{n\times k}$ and $Y \in \mathbb{O}^{m\times k}$ have orthonormal columns, where $1 \le k < \min\{m, n\}$ (usually $k \ll \min\{m, n\}$). Then we immediately have the following equivalent reformulation of OCCA (5):

$$\max_{X\in\mathbb{O}^{n\times k},Y\in\mathbb{O}^{m\times k}} \left\{F(X,Y) := \frac{\mathrm{tr}^2(X^{\mathrm{T}}CY)}{\mathrm{tr}(X^{\mathrm{T}}AX)\,\mathrm{tr}(Y^{\mathrm{T}}BY)}\right\}, \tag{10a}$$

$$\text{s.t. } \mathrm{tr}(X^{\mathrm{T}}CY) \geq 0. \tag{10b}$$

In the next subsection we will present an algorithm to solve (10). Our algorithm can take the advantage of the specific structure of the problem with theoretical guarantees of convergence as shown in Section 4. Furthermore, in Section 5, we show that the algorithm can be extended easily to handle an orthogonal multiview CCA model.

3.2 The proposed algorithm

We propose the numerical scheme as shown in Algorithm 1 by maximizing $F(X,Y)$ (or equivalently $f(X,Y)$) alternatively with respective to X and Y until convergence.

Algorithm 1 An alternating optimization scheme for (10)

Input: $\{X^{(0)},Y^{(0)}\}$ with $X^{(0)}\in\mathbb{O}^{n\times k}, Y^{(0)}\in\mathbb{O}^{m\times k}$.
Output: a solution $\{X^{(\nu)},Y^{(\nu)}\}$ to (10).
1: **for** $\nu = 1,2,\ldots$ until convergence **do**
2: solve, subject to $\mathrm{tr}(X^{\mathrm{T}}CY^{(\nu-1)})\geq 0$,

$$X^{(\nu)} \in \arg\max_{X\in\mathbb{O}^{n\times k}} F(X,Y^{(\nu-1)}); \tag{11}$$

3: solve, subject to $\mathrm{tr}([X^{(\nu)}]^{\mathrm{T}}CY)\geq 0$,

$$Y^{(\nu)} \in \arg\max_{Y\in\mathbb{O}^{m\times k}} F(X^{(\nu)},Y); \tag{12}$$

4: compute SVD of $(X^{(\nu)})^{\mathrm{T}}CY^{(\nu)} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^{\mathrm{T}}$;
5: set $X^{(\nu)}\leftarrow X^{(\nu)}\widetilde{U}$ and $Y^{(\nu)}\leftarrow Y^{(\nu)}\widetilde{V}$;
6: **end for**
7: **return** $\{X^{(\nu)},Y^{(\nu)}\}$.

The role of line 4 in Algorithm 1 is to make sure $X^{(\nu)}$ and $Y^{(\nu)}$ are always well aligned. It is based on the structure of the function $F(X,Y)$: Given a pair $(X^{(\nu)},Y^{(\nu)})$, the denominator is unchanged when this pair is changed to $(X^{(\nu)}U,Y^{(\nu)}V)$ for any $U,V\in\mathbb{O}^{k\times k}$, while the numerator is maximized by the particular pair $(U,V)=(\widetilde{U},\widetilde{V})$ given by

$$(\widetilde{U},\widetilde{V}) = \arg\max_{U,V\in\mathbb{O}^{k\times k}} \mathrm{tr}\left(U^{\mathrm{T}}(X^{(\nu)})^{\mathrm{T}}CY^{(\nu)}V\right),$$

as can be justified by Lemma 3 in Subsection 3.3. The maximum value is $\sum_{i=1}^{k}\sigma_i\left((X^{(\nu)})^{\mathrm{T}}CY^{(\nu)}\right)$. Stopping criteria for line 1 will be discussed later in Subsection 6.2.

The efficiency of Algorithm 1 relies heavily on solving the sub-maximization problems (11) and (12). Abstractly, they are of the following type

$$\max_{G\in\mathbb{O}^{n\times k}} \eta(G) := \frac{\mathrm{tr}^2(G^{\mathrm{T}}D)}{\mathrm{tr}(G^{\mathrm{T}}AG)}, \tag{13}$$

subject to $\mathrm{tr}(G^{\mathrm{T}}D)\geq 0$, where $0\neq D\in\mathbb{R}^{n\times k}$ and $A\succ 0$. In Subsection 3.3, we present an SCF iteration that directly aims at solving (13).

3.3 A novel algorithm for solving (13)

It can be seen that the global maximum of (13) is positive unless $D=0$. Moreover, (13) is very much like the trace ratio (or trace quotient) maximization, i.e., maximizing $\mathrm{tr}(G^{\mathrm{T}}A_1G)/\mathrm{tr}(G^{\mathrm{T}}A_2G)$ over $G\in\mathbb{O}^{n\times k}$ with given $A_1,A_2\succ 0$, for which an efficient SCF iteration is available [18], [19], [20], [21]. It has been proved that the SCF iteration is globally convergent and the convergence is locally quadratic. Historically, the SCF iteration was commonly used to solve the Eigenvector-Dependent Nonlinear Eigenvalue Problem (NEPv) [18] from the Kohn–Sham density functional theory in electronic structure calculations [22], [23]. Recently, it has been attracting a great deal of attention in data science (e.g., [12], [13], [18], [24], [25]).

Next, we will first transform the problem (13) into a novel NEPv that is not quite the same as the KKT condition of (13), and then apply the SCF iteration to solve the NEPv. The most challenging part is the convergence analysis of the resulting SCF iteration, which will be studied in Section 4.

3.3.1 A nonlinear eigenvalue problem

We will first derive the partial derivative $\partial\eta(G)/\partial G$, where all entries of G are treated as independent variables. Consequently, the gradient $\mathrm{grad}\,\eta(G)$ at $G\in\mathbb{O}^{n\times k}$ on the Stiefel manifold $\mathbb{O}^{n\times k}$ is given by

$$\mathrm{grad}\,\eta(G) = \Pi_G\left(\frac{\partial\eta(G)}{\partial G}\right)\in\mathcal{T}_G\mathbb{O}^{n\times k}, \tag{14}$$

where $\Pi_G(Z) = Z - G\,\mathrm{sym}(G^{\mathrm{T}}Z)$ for $Z\in\mathbb{R}^{n\times k}$, see e.g., [9].

By straightforward calculations, we have

$$\frac{\partial\eta(G)}{\partial G} = \frac{2\,\mathrm{tr}(G^{\mathrm{T}}D)}{\mathrm{tr}(G^{\mathrm{T}}AG)}D - \frac{2\,\mathrm{tr}^2(G^{\mathrm{T}}D)}{\mathrm{tr}^2(G^{\mathrm{T}}AG)}AG.$$

and

$$\mathrm{grad}\,\eta(G) = -\frac{2}{\xi^2(G)}\Big\{[AG-\xi(G)D]-GM(G)\Big\}, \tag{15}$$

where

$$\xi(G) = \frac{\mathrm{tr}(G^{\mathrm{T}}AG)}{\mathrm{tr}(G^{\mathrm{T}}D)},\quad M(G) = \mathrm{sym}(G^{\mathrm{T}}AG - \xi(G)G^{\mathrm{T}}D). \tag{16}$$

From (15), we immediately have Lemma 1 below.

Lemma 1. *If G is a KKT point of* (13)*, then*

$$AG - \xi(G)D = GM(G). \tag{17}$$

Note that the condition (17) is a type of nonlinear Sylvester equation but with the orthogonality constraint $G^{\mathrm{T}}G = I_k$. To solve it, we will convert it into an NEPv so that the SCF iteration is applicable. One straightforward way is to use the constraint $G^{\mathrm{T}}G=I_k$ and then rewrite (17) equivalently as $[A-\xi(G)DG^{\mathrm{T}}]G = GM(G)$. However, we notice that the matrix $A-\xi(G)DG^{\mathrm{T}}$ is not necessarily symmetric, even at a maximizer G. This means that we cannot ensure $A-\xi(G)DG^{\mathrm{T}}$ has real eigenvalues at $G\in\mathbb{O}^{n\times k}$. To overcome that obstacle, we construct the following NEPv

$$E(G)G = G\widehat{M}(G), \tag{18}$$

where $\widehat{M}(G) = G^{\mathrm{T}}E(G)G\in\mathbb{R}^{k\times k}$ and

$$E(G) := A - \xi(G)(DG^{\mathrm{T}} + GD^{\mathrm{T}}). \tag{19}$$

Evidently, $E(G)$ is always symmetric. The following lemma establishes a relation between (17) and (18).

Lemma 2. *Suppose $G\in\mathbb{O}^{n\times k}$. Then G satisfies* (17) *if and only if G is an eigenbasis matrix of $E(G)$, i.e., G satisfies* (18)*.*

Lemma 2 characterizes any maximizer G of (13) as an orthonormal eigenbasis matrix of $E(G)$. By (18), we find

$$\mathrm{eig}(\widehat{M}(G)) = \{\lambda_{\pi_1}(E(G)), \ldots, \lambda_{\pi_k}(E(G))\} \subseteq \mathrm{eig}(E(G)),$$

where $\{\pi_1 \leq \cdots \leq \pi_k\} \subset \{1, 2, \ldots, n\}$.

3.3.2 Eigenspace associated with a global maximizer

Even though our maximization problem (13) is very much like the trace ratio problem [19], unfortunately, it does not enjoy some nice properties as the trace ratio problem. For example, it is shown that any local maximizer of the trace ratio problem is also a global solution. The problem (13) in general admits local but non-global maximizers (see Example A in the supplementary material). The following theorem provides a necessary condition for the local maximizer in terms of the NEPv in (18).

Theorem 1. *If G is a local maximizer of* (13)*, then $\mathcal{R}(G)$ is an eigenspace of $E(G)$ associated with eigenvalues $\lambda_{\pi_1}(E(G)) \leq \cdots \leq \lambda_{\pi_k}(E(G))$ satisfying $\pi_1 \leq k$.*

Theorem 1 indicates that for any local maximizer G, the smallest eigenvalue associated with the eigenspace $\mathcal{R}(G)$ must be no bigger than $\lambda_k(E(G))$. This offers a necessary condition for a KKT point to be a local maximizer. As a much stronger version, the next theorem says that any global maximizer G must be an eigenbasis matrix associated with the k smallest eigenvalues of $E(G)$.

Theorem 2. *If G_{opt} is a global maximizer of* (13)*, then G_{opt} is an orthonormal eigenbasis matrix associated with the k smallest eigenvalues of $E(G_{\mathrm{opt}})$. Moreover, the matrix $G_{\mathrm{opt}}^{\mathrm{T}} D$ is symmetric and positive semidefinite.*

3.3.3 A self-consistent-field (SCF) iteration

Suppose G_{opt} is a global maximizer of (13). NEPv (18), equipped with the necessary condition in Theorem 2, implicitly defines a fixed point mapping which maps the eigenspace $\mathcal{R}(G_{\mathrm{opt}})$ associated with the k smallest eigenvalues of $E(G_{\mathrm{opt}})$ to itself. To find this eigenspace numerically, the SCF iteration is a natural technique which is outlined in Algorithm 2.

Algorithm 2 A SCF iteration for solving (13)

Input: $G_{(0)} \in \mathbb{O}^{n\times k}$;
Output: approximate maximizer G to (13).
1: **for** $\nu = 1, 2, \ldots$ until convergence **do**
2: construct $E_{(\nu)} = E(G_{(\nu-1)})$ as in (18);
3: compute an orthonormal eigenbasis matrix $G_{(\nu)}$ associated with the k smallest eigenvalues of $E_{(\nu)}$;
4: compute SVD: $G_{(\nu)}^{\mathrm{T}} D = U\Sigma V^{\mathrm{T}}$;
5: update $G_{(\nu)} \leftarrow G_{(\nu)} UV^{\mathrm{T}}$;
6: **end for**
7: **return** $G_{(\nu)}$.

Remark 1. We have three comments for Algorithm 2.

(a) Comparing with the standard SCF iteration [18] for a general NEPv, our proposed SCF version for (18) has an additional step at lines 4 and 5, which aims to maximally push up the value of objective function η in (13) for an (arbitrarily) chosen orthonormal eigenbasis matrix $G_{(\nu)} \in \mathbb{O}^{n\times k}$ of $E(G_{(\nu-1)})$ associated with its k smallest eigenvalues. We note that an eigenbasis matrix is not unique. In fact, $\widehat{G}_{(\nu)} = G_{(\nu)} P$ for any $P \in \mathbb{O}^{k\times k}$ is also one. Since $\mathrm{tr}(\widehat{G}_{(\nu)}^{\mathrm{T}} A \widehat{G}_{(\nu)}) \equiv \mathrm{tr}(G_{(\nu)}^{\mathrm{T}} A G_{(\nu)})$ but $\mathrm{tr}(\widehat{G}_{(\nu)}^{\mathrm{T}} D) \neq \mathrm{tr}(G_{(\nu)}^{\mathrm{T}} D)$ in general, it makes sense to update $G_{(\nu)}$ to $\widehat{G}_{(\nu)}$ so that $\mathrm{tr}(\widehat{G}_{(\nu)}^{\mathrm{T}} D)$ is maximized over $P \in \mathbb{O}^{k\times k}$. That is when Lemma 3 below comes to help.

(b) The goal of Algorithm 2 is to seek a maximizer of (13) and at a maximizer G, $\mathrm{grad}\,\eta(G) = 0$ in theory. Considering roundoff errors in evaluating $\mathrm{grad}\,\eta$ according to (15), a reasonable stopping criterion to use at line 1 of Algorithm 2 is

$$\xi^{-2}(G_{(\nu)}) \frac{\|\mathrm{grad}\,\eta(G_{(\nu)})\|_1}{\|A\|_1 + \xi(G_{(\nu)})\|D\|_1} \leq \epsilon_{\mathrm{scf}},$$

where ϵ_{scf} is a preset tolerance. In our later experiments, we use $\epsilon_{\mathrm{scf}} = 10^{-5}$ and as a safe guard, we set 30 as the maximum number of iterations allowed. Here $\|\cdot\|_1$ is the ℓ_1-matrix norm which works equally well for all practical purposes as to the matrix-spectral norm that we should use ideally but the latter is more expensive to compute. A good initial guess $G_{(0)}$ is the orthogonal factor in the polar decomposition of D, which maximizes the numerator of $\eta(G)$ among all G such that $\mathcal{R}(G) = \mathcal{R}(D)$. (An additional modification is required when $\mathrm{rank}(D) < k$.)

(c) At line 3, we need to compute an orthonormal eigenbasis matrix $G_{(\nu)}$ of $E_{(\nu)}$ which is $n \times n$. For modest n, say up to a few hundred, we may simply call, e.g., MATLAB's `eig`, to compute a full eigendecomposition of $E_{(\nu)}$ which costs $O(n^3)$ flops, but for large n, $O(n^3)$ is too much and we should use an iterative eigensolver. More discussions will come later.

Lemma 3. *Let $W \in \mathbb{R}^{k\times k}$. Then $|\mathrm{tr}(W)| \leq \sum_{i=1}^k \sigma_i(W)$. If $|\mathrm{tr}(W)| = \sum_{i=1}^k \sigma_i(W)$, then W is symmetric and is either positive semidefinite when $\mathrm{tr}(W) \geq 0$, or negative semidefinite when $\mathrm{tr}(W) \leq 0$.*

According to Lemma 3, lines 4 and 5 of Algorithm 2 ensures

$$\mathrm{tr}(G_{(\nu)}^{\mathrm{T}} D) = \max_{P\in\mathbb{O}^{k\times k}} \mathrm{tr}((G_{(\nu)}P)^{\mathrm{T}} D) = \sum_{i=1}^k \sigma_i(G_{(\nu)}^{\mathrm{T}} D) \geq 0,$$

and also $G_{(\nu)}^{\mathrm{T}} D \succeq 0$ for all $\nu \geq 1$.

4 Main Theoretical Results

4.1 Brief sketch

In this section, we shall establish the convergence of both the outer-loop alternating optimization scheme of Algorithm 1 and the inner core SCF iteration of Algorithm 2 for the subproblem (13). Our analysis reveals that the inner SCF iteration converges monotonically (Theorem 4(ii)), and generally, any limit point is a KKT point satisfying the necessary optimality condition given in Theorem 2 for the global maximizer (Theorem 4(v)). Moreover, the linear convergence rate of the SCF iteration is discussed (Theorem 5). Based

on the inner SCF solver for the subproblem (13), the monotonic convergence of the outer-loop alternating optimization scheme of Algorithm 1 is guaranteed (Theorem 3).

4.2 Analysis for Algorithm 1

We first mention that our specialized SCF iteration Algorithm 2 with the additional procedure in lines 4 and 5 brings another nice property for the sequence $\{(X^{(\nu)}, Y^{(\nu)})\}_{\nu=1}^{\infty}$, that is, $(X^{(\nu)})^{\mathrm{T}} C Y^{(\nu)}$ is symmetric and positive semidefinite, which is a necessary condition for any global solution pair $(X_{\mathrm{opt}}, Y_{\mathrm{opt}})$ (see Theorem 3(i)). Moreover, by using the effective solvers for (11) and (12), Algorithm 1 always converges. We summarize these results in the following theorem.

Theorem 3. *Let $(X_{\mathrm{opt}}, Y_{\mathrm{opt}})$ be an optimal solution pair to* (10) *and $(X^{(\nu)}, Y^{(\nu)})$ be the ν-th approximation by Algorithm 1. Then*

(i) *$X_{\mathrm{opt}}^{\mathrm{T}} C Y_{\mathrm{opt}}$ is symmetric and positive semidefinite.*
(ii) *$(X^{(\nu)})^{\mathrm{T}} C Y^{(\nu)}$ is symmetric and positive semidefinite for $\nu \geq 1$, and thus for any limit pair (X, Y) of $\{(X^{(\nu)}, Y^{(\nu)})\}_{\nu=1}^{\infty}$, $X^{\mathrm{T}} C Y$ is symmetric and positive semidefinite.*
(iii) *The sequence $\{F(X^{(\nu)}, Y^{(\nu)})\}_{\nu=1}^{\infty}$ is monotonically increasing and converges.*

4.3 Analysis for Algorithm 2

Before our discussion on the convergence of the SCF iteration (Algorithm 2), we first provide the following two lemmas.

Lemma 4. *For any $G, \widehat{G} \in \mathbb{O}^{n\times k}$, if $\widehat{G}^{\mathrm{T}} D = D^{\mathrm{T}} \widehat{G} \succeq 0$ and*

$$\mathrm{tr}(\widehat{G}^{\mathrm{T}} E(G) \widehat{G}) \leq \mathrm{tr}(G^{\mathrm{T}} E(G) G), \tag{20}$$

then $\eta(\widehat{G}) \geq \eta(G)$. Furthermore, if the inequality in (20) *is strict, then $\eta(\widehat{G}) > \eta(G)$.*

The action at line 3 of Algorithm 2 can now be justified by Lemma 4. In fact, the chosen $G_{(\nu)}$ satisfies

$$\mathrm{tr}(G_{(\nu)}^{\mathrm{T}} E(G_{(\nu-1)}) G_{(\nu)}) \leq \mathrm{tr}(G_{(\nu-1)}^{\mathrm{T}} E(G_{(\nu-1)}) G_{(\nu-1)})$$

and thus $\eta(G_{(\nu)}) \geq \eta(G_{(\nu-1)})$, implying monotonic increase of $\{\eta(G_{(\nu)})\}$.

As an eigenbasis matrix is not unique, one may ask if $G_{(\nu)}$ at lines 4 and 5 of Algorithm 2 is well-defined. The next lemma addresses this issue.

Lemma 5. *At line 3 of Algorithm 2, if the eigenvalue gap*

$$\zeta_{\nu-1} = \lambda_{k+1}(E(G_{(\nu-1)})) - \lambda_k(E(G_{(\nu-1)})) > 0,$$

then any two orthonormal eigenbasis matrices $\widehat{G}_{(\nu)}$ and $\widetilde{G}_{(\nu)}$ associated with the k smallest eigenvalues of $E(G_{(\nu-1)})$ satisfy $\widetilde{G}_{(\nu)} = \widehat{G}_{(\nu)} Q$ for some orthogonal matrix $Q \in \mathbb{O}^{k\times k}$. Furthermore, if additionally $\mathrm{rank}(D^{\mathrm{T}} \widehat{G}_{(\nu)}) = k$, then the next approximation $G_{(\nu)}$ from line 4 of Algorithm 2 is uniquely determined.

We next provide basic convergence properties of the SCF iteration Algorithm 2 for solving the problem (13).

Theorem 4. *Let the sequence $\{G_{(\nu)}\}$ be generated by the SCF iteration (Algorithm 2). Then*

(i) *For each $\nu \geq 1$, $D^{\mathrm{T}} G_{(\nu)} \succeq 0$ and $\mathrm{tr}(G_{(\nu)}^{\mathrm{T}} D) = \sum_{j=1}^{k} \sigma_j(G_{(\nu)}^{\mathrm{T}} D)$;*
(ii) *The sequence $\{\eta(G_{(\nu)})\}$ is monotonically increasing and convergent;*
(iii) *If*

$$\mathrm{tr}(G_{(\nu)}^{\mathrm{T}} E(G_{(\nu-1)}) G_{(\nu)}) < \mathrm{tr}(G_{(\nu-1)}^{\mathrm{T}} E(G_{(\nu-1)}) G_{(\nu-1)}), \tag{21}$$

then $\eta(G_{(\nu-1)}) < \eta(G_{(\nu)})$;
(iv) *$\{G_{(\nu)}\}$ has a convergent subsequence $\{G_{(\nu)}\}_{\nu\in\mathcal{I}}$;*
(v) *Let $\{G_{(\nu)}\}_{\nu\in\mathcal{I}}$ be any convergent subsequence of $\{G_{(\nu)}\}$ with the accumulation point G_* satisfying*

$$\zeta = \lambda_{k+1}(E(G_*)) - \lambda_k(E(G_*)) > 0. \tag{22}$$

Then G_* satisfies the first order optimality condition in Lemma 1 and also the necessary condition for a global maximizer in Theorem 2.

Remark 2. We have three comments for Theorem 4.

(a) Item (iii) of Theorem 4 implies that, to only guarantee monotonicity of $\{\eta(G_{(\nu)})\}$, the partial eigendecomposition at line 3 of Algorithm 2 can be inexact. In particular, we can choose any approximation $G_{(\nu)} \in \mathbb{O}^{n\times k}$ satisfying (21), and then refine it by line 4 to ensure $D^{\mathrm{T}} G_{(\nu)} \succeq 0$; by Lemma 4, $\eta(G_{(\nu)}) > \eta(G_{(\nu-1)})$ still holds. This facilitates us to employ sophisticated eigensolvers [26] for the computation task at line 3.
(b) Item (iv) is rather obvious because $\{G_{(\nu)}\}$ is a bounded sequence in $\mathbb{R}^{n\times k}$. It is explicitly listed to substantiate part of the assumption in item (v). A stronger claim in Theorem 5 later says the entire sequence $\{G_{(\nu)}\}$ converges under a mild condition.
(c) Item (v) shows one of advantages of our SCF iteration over the generic Riemannian optimization methods for solving the core subproblem (13). In particular, as our SCF iteration is built upon the necessary conditions of a global maximizer, besides the general KKT conditions, the convergent point also fulfills the necessary conditions for being a global maximizer.

To further analyze the convergence of the sequence $\{G_{(\nu)}\}$, we now consider the sequence $\{\mathcal{R}(G_{(\nu)})\}$ of subspaces. For this purpose, we denote by $\|\cdot\|_{\mathrm{ui}}$ any unitarily invariant norm [27], and introduce the distance measure between two subspaces $\mathcal{G}$ and $\mathcal{Y}$ of dimension k [28, p.95]

$$\mathrm{dist}_{\mathrm{ui}}(\mathcal{G}, \mathcal{Y}) := \|\sin\Theta(\mathcal{G}, \mathcal{Y})\|_{\mathrm{ui}} \tag{23}$$

in terms of the matrix of the canonical angles between $\mathcal{G}$ and $\mathcal{Y}$:

$$\Theta(\mathcal{G}, \mathcal{Y}) = \mathrm{Diag}(\theta_1(\mathcal{G}, \mathcal{Y}), \ldots, \theta_k(\mathcal{G}, \mathcal{Y})).$$

Let $\mathcal{G} = \mathcal{R}(G)$ and $\mathcal{Y} = \mathcal{R}(Y)$, where $G, Y \in \mathbb{R}^{n\times k}$ with $G^{\mathrm{T}} G = Y^{\mathrm{T}} Y = I_k$. The canonical angles $\theta_1(\mathcal{G}, \mathcal{Y}) \geq \cdots \geq \theta_1(\mathcal{G}, \mathcal{Y})$ are defined by

$$0 \leq \theta_i(\mathcal{G}, \mathcal{Y}) := \arccos\sigma_i(G^{\mathrm{T}} Y) \leq \frac{\pi}{2} \quad \text{for } 1 \leq i \leq k.$$

The collection of all k-dimensional subspaces in $\mathbb{R}^n$ is the so-called Grassmann manifold $\mathscr{G}_k(\mathbb{R}^n)$, and the distance measure (23) is a unitarily invariant metric [28, p.95] on

$\mathscr{G}_k(\mathbb{R}^n)$. For the trace norm, also known as the nuclear norm, we have

$$\mathrm{dist}_{\mathrm{tr}}(\mathcal{G},\mathcal{Y}) = \sum_{j=1}^{k} \sin\theta_j(\mathcal{G},\mathcal{Y}).$$

Using the metric $\mathrm{dist}_{\mathrm{tr}}(\mathcal{G},\mathcal{Y})$, we have the following convergence result for the sequence $\{G_{(\nu)}\}$ by the SCF iteration in Algorithm 2.

Theorem 5. *Let the sequence $\{G_{(\nu)}\}$ be generated by the SCF iteration (Algorithm 2), and let G_* be an accumulation point of $\{G_{(\nu)}\}$. Suppose that $\mathcal{R}(G_*)$ is an isolated accumulation point (in the metric* (23)*) of $\{\mathcal{R}(G_{(\nu)})\}_{\nu=0}^{\infty}$. Then*

(i) *$\{\mathcal{R}(G_{(\nu)})\}_{\nu=0}^{\infty}$ converges to $\mathcal{R}(G_*)$;*

(ii) *if also $\mathrm{rank}(G_*^{\mathrm{T}}D) = k$ and if* (22) *holds, then $\{G_{(\nu)}\}_{\nu=0}^{\infty}$ converges to G_* (in the standard Euclidean metric), and for sufficiently large ν,*

$$\mathrm{dist}_{\mathrm{tr}}(\mathcal{R}(G_*),\mathcal{R}(G_{(\nu+1)})) \le c_0\|G_{(\nu)} - G_*\|_{\mathrm{tr}}, \quad (24)$$

where

$$c_0 = \frac{3\|D\|_2}{\zeta}\left(\sqrt{\frac{\|A\|_{(k)}}{\eta(G_*)}} + 2k\frac{\|A\|_2 + \sqrt{\frac{\|A\|_{(k)}}{\eta(G_*)}}\|D\|_2}{\sqrt{\eta(G_*)\omega_k(A)}}\right),$$

with $\|A\|_{(k)} = \sum_{j=1}^{k}\sigma_j(A)$, $\omega_k(A) = \sum_{j=1}^{k}\sigma_{n-j+1}(A)$.

What is remarkable about Theorem 5 is that we start with an accumulation point G_* which always exists because $\mathbb{O}^{n\times k}$ is a bounded set in $\mathbb{R}^{n\times k}$ and thus is compact, and end up with the conclusions that $\{\mathcal{R}(G_{(\nu)})\}_{\nu=0}^{\infty}$ converges to $\mathcal{R}(G_*)$ and that $\{G_{(\nu)}\}_{\nu=0}^{\infty}$ converges to G_* under mild conditions.

5 Algorithmic Extension for OMCCA

We propose a new formulation of OMCCA and then solve it by extending our algorithms in Section 3.

Analogously to (5), our new formulation of OMCCA naturally arises:

$$\max_{\{X_i\in\mathbb{O}^{n_i\times k}\}} f(\{X_i\}). \quad (25)$$

where $1 \le k \le \min\{n_1,\ldots,n_\ell,q\}$, and

$$f(\{X_i\}) = \sum_{\substack{i,\,j=1\\ i\neq j}}^{\ell}\rho_{ij}\frac{\mathrm{tr}(X_i^{\mathrm{T}}C_{i,j}X_j)}{\sqrt{\mathrm{tr}(X_i^{\mathrm{T}}C_{i,i}X_i)}\sqrt{\mathrm{tr}(X_j^{\mathrm{T}}C_{j,j}X_j)}} \quad (26)$$

with weighting factors $\rho_{ij} \ge 0$. Ideally, the optimal weights should be learned from data, but this is out of the scope of this paper. Here, we employ some heuristic weighting schemes. To begin with, we define

$$\widehat{\rho}_{ij} = \frac{\sum_{r=1}^{\mathrm{rank}(C_{i,j})}\sigma_r(C_{i,j})}{\sqrt{\mathrm{tr}(C_{i,i})\,\mathrm{tr}(C_{j,j})}}, \ \text{for } i,j = 1,\ldots,\ell. \quad (27)$$

It is known $0 \le \widehat{\rho}_{ij} \le 1$ [29, (3.5.22) on p.212]. Envision a graph of ℓ nodes corresponding to datasets S_i, respectively, with every two nodes connected with an edge whose weight is to be determined. We take three heuristic strategies:

1) uniform weighting: $\rho_{ij} = 1, \forall\, i,j = 1,\ldots,\ell$.
2) tree weighting: find the minimal spanning tree of the graph with the edge (i,j) having weight $1-\widehat{\rho}_{ij}$, record the spanning tree with its edge weights reset back to $\widehat{\rho}_{ij}$ and weights $\widehat{\rho}_{ij}$ for all other edges not in the tree reset to 0.
3) top-p weighting: find the p largest weights among $\widehat{\rho}_{ij}$ for $i > j$, and reset all other weights $\widehat{\rho}_{ij}$ to 0.

For the last two strategies, we apply the soft-max function over those reset weights $\widehat{\rho}_{ij}$ with a bandwidth parameter (e.g., 20 used in our experiments) to yield ρ_{ij} to use in (26). As a by-product, the sum of all ρ_{ij} is 1.

Based on the machinery we have built in Section 3, we propose to optimize $f(\{X_i\})$ cyclically over each matrix variable X_i in the styles similar to either the Jacobi or the Gauss-Seidel iteration for the linear system of equations [30]. Specifically, we establish an inner-outer iterative method to solve (25). The most outer iteration – each step called a cycle – generates from the current approximation $\{X_i^{(\nu)}\}_{i=1}^{\ell}$ to the next $\{X_i^{(\nu+1)}\}_{i=1}^{\ell}$ of the maximizer set of (25); each cycle can be of an either the Jacobi-style or Gauss-Seidel-style updating scheme that relies on the proposed novel SCF iteration for solving a series of subproblems in the form of (13).

Some denominators in f in (26) may vanish if $\mathrm{rank}(C_{i,i})+k \le n_i$, which is possible when $q \ll n_i$ for some i. When it does happen, numerical difficulties may arise. To circumvent them, we propose to add range constraints

$$\mathcal{R}(X_i) \subset \mathcal{R}(S_i) \quad \text{for } i = 1,2,\ldots,\ell. \quad (28)$$

In what follows, we describe an SVD-based implementation. Let the SVDs of S_i be

$$S_i = U_i\Sigma_i V_i^{\mathrm{T}},\ U_i \in \mathbb{R}^{n_i\times r_i},\ V_i \in \mathbb{R}^{q\times r_i},\ \Sigma_i \in \mathbb{R}^{r_i\times r_i}, \quad (29)$$

where $r_i = \mathrm{rank}(S_i)$. With the SVDs in (29), we have

$$X_i^{\mathrm{T}}S_iS_j^{\mathrm{T}}X_j = X_i^{\mathrm{T}}U_i\Sigma_iV_i^{\mathrm{T}}V_j\Sigma_jU_j^{\mathrm{T}}X_j =: \widehat{X}_i^{\mathrm{T}}\Sigma_iV_i^{\mathrm{T}}V_j\Sigma_j\widehat{X}_j,$$

where $\widehat{X}_i = U_i^{\mathrm{T}}X_i \in \mathbb{R}^{r_i\times k}$. Under (28), we will have $X_i = U_i\widehat{X}_i$. The function $f(\{X_i\})$ is then transformed into

$$\sum_{i\neq j}\rho_{ij}\frac{\mathrm{tr}(\widehat{X}_i^{\mathrm{T}}\Sigma_iV_i^{\mathrm{T}}V_j\Sigma_j\widehat{X}_j)}{\sqrt{\mathrm{tr}(\widehat{X}_i^{\mathrm{T}}\Sigma_i^2\widehat{X}_i)}\sqrt{\mathrm{tr}(\widehat{X}_j^{\mathrm{T}}\Sigma_j^2\widehat{X}_j)}} =: g(\{\widehat{X}_i\}),$$

and

$$\max_{X_i\in\mathbb{O}^{n_i\times k},\,\mathcal{R}(X_i)\subset\mathcal{R}(S_i),\,\forall i} f(\{X_i\}) = \max_{\widehat{X}_i\in\mathbb{O}^{r_i\times k},\,\forall i} g(\{\widehat{X}_i\}).$$

The key step to maximize $g(\{\widehat{X}_i\})$ by either the Jacobi- or Gauss-Seidel-style updating scheme is to maximize it, for any $s \in \{1,\cdots,\ell\}$, over $\widehat{X}_s$ while freezing all other $\widehat{X}_j$ $(j \neq s)$. That is equivalent to

$$\max_{\widehat{X}_s\in\mathbb{O}^{n_s\times k}} \frac{\mathrm{tr}(\widehat{X}_s^{\mathrm{T}}D_s)}{\sqrt{\mathrm{tr}(\widehat{X}_s^{\mathrm{T}}\Sigma_s^2\widehat{X}_s)}}, \quad (30)$$

where

$$D_s(\{\widehat{X}_i\}_{i\neq s}) = \Sigma_sV_s^{\mathrm{T}}\sum_{j\neq s}\rho_{sj}\frac{V_j\Sigma_j\widehat{X}_j}{\sqrt{\mathrm{tr}(\widehat{X}_j^{\mathrm{T}}\Sigma_j^2\widehat{X}_j)}}. \quad (31)$$

Problem (30) is equivalent to

$$\max_{\widehat{X}_s\in\mathbb{O}^{n_s\times k}} \frac{\mathrm{tr}^2(\widehat{X}_s^{\mathrm{T}}D_s)}{\mathrm{tr}(\widehat{X}_s^{\mathrm{T}}\Sigma_s^2\widehat{X}_s)}, \quad (32)$$

subject to $\mathrm{tr}(\widehat{X}_s^{\mathrm{T}}D_s) \ge 0$, which takes the same form as (13), and has been studied in Subsection 3.3.

For the ease of reference, we name the above proposed extension algorithm for OMCCA as Range Constrained OMCC (RCOMCCA). More details are presented in part III of the supplementary material. It is worth noting that RCOMCCA allows two updating schemes and is capable of integrating various weighting schemes. Hence, we name the variants of RCOMCCA by suffixing "-G" for Gauss-Seidel-style and "-J" for the Jacobi-style, together with three weighting schemes shown in bracket. As a result, there are six variants of RCOMCCA in total (as listed in the first column of Table 4 in Section 6).

6 Experiments

6.1 Implementation details and complexity

First we note that the function f in (4) is well-defined when its denominator never vanishes. This is guaranteed if $\mathrm{rank}(C_{1,1}) + k > n$ and $\mathrm{rank}(C_{2,2}) + k > m$. The same can be said about the function F in (10). Otherwise, numerical difficulty may arise, and some kind of pre-cautionary measure such as the range constraint discussed in Section 5 must be taken. For simplicity, we will focus our discussion here on the case without such a pre-cautionary measure (which, in fact, is not needed for all datasets used in this section, except yeast_ribosomal in Table 3). Indeed, our discussion can be minorly modified if there is one.

Our SCF-based algorithm for solving OCCA (10), referred as OCCA-scf for short hereafter, is Algorithm 1 that uses Algorithm 2 as its workhorse to solve all involved subproblems in the form of (13). Current implementation for line 3 of Algorithm 2, when $n \leq 500$, calls MATLAB's mex version `mexeig` of LAPACK's [31] eigen-decomposition subroutine `dsyevd`[2] to compute $G_{(\nu)}$. For $n > 500$, it uses the locally optimal block preconditioned conjugate gradient method[3] (LOBPCG) (see [32], [33]) with the diagonal preconditioner. As LOBPCG searches an approximate $G_{(\nu)}$ by optimizing the Rayleigh quotient initially in a subspace containing $\mathcal{R}(G_{(\nu-1)})$, the condition (21) is always met, implying that the sequence $\{\eta(G_{(\nu)})\}$ is monotonically increasing and convergent (Theorem 4).

To get an idea of how the overall computational complexity in flops is, we let n_{alt} be the number of full alternating iterations taken by Algorithm 1 and n_{scf} be the average number of SCF iterative steps taken by Algorithm 2. The overall complexity is roughly

$$n_{\mathrm{alt}} n_{\mathrm{scf}} \left[\mathrm{cost}_{\mathrm{eig}} + O(nk^2 + mk^2 + k^3)\right],$$

where $O(nk^2 + mk^2 + k^3)$ is for the SVD and updating at lines 4 and 5 of Algorithm 2, and $\mathrm{cost}_{\mathrm{eig}}$ is the cost for executing its line 3. For using full eigen-decomposition such as `dsyevd`, $\mathrm{cost}_{\mathrm{eig}} = O(n^3 + m^3)$, but for using an iterative solver such as LOBPCG, $\mathrm{cost}_{\mathrm{eig}} = O(nk^2 + mk^2)$. Both n_{alt} and n_{scf} are capped at 30. In the case when $n \approx m \approx q$, overall computational complexity is $O(n^3)$ for $n \leq 500$ and $O(nk^2)$ for $n > 500$ for the current implementation.

2. `mexeig` (available at: `www.math.nus.edu.sg/~matsundf/`) is a MATLAB interface to call LAPACK eigen-decomposition subroutine `dsyevd` of a real symmetric matrix.

3. The MATLAB version of LOBPCG is available at: http://cn.mathworks.com/matlabcentral/fileexchange/48-lobpcg-m.

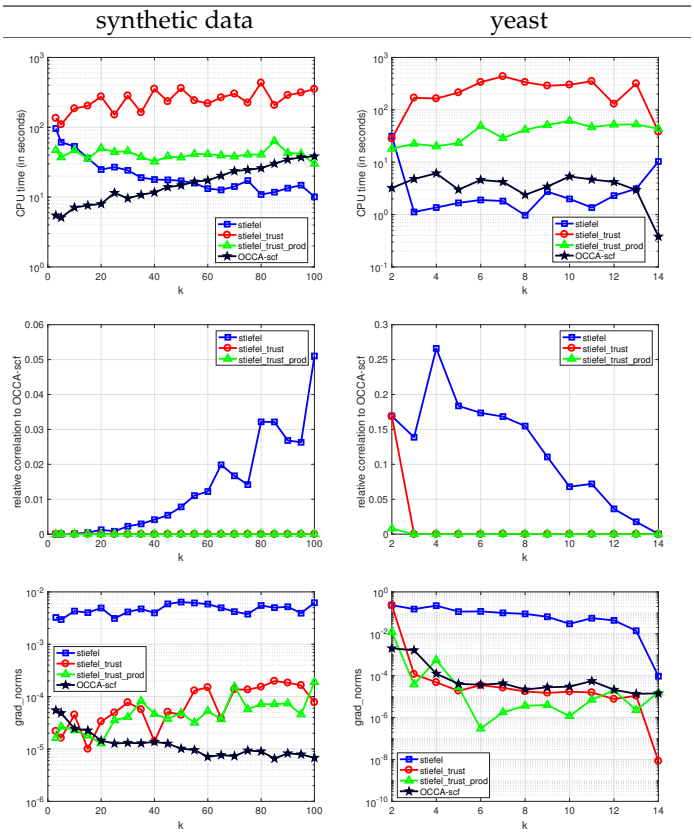


Fig. 1. Comparison of four optimization methods on synthetic data and multi-label classification data yeast in terms of three different criteria. The plots in the second row are for the differences: subtracting the objective value by each of the three stiefel methods from the one by OCCA-scf.

6.2 Comparisons with generic optimization methods

We conduct extensive experiments to compare OCCA-scf with three generic optimization methods over Stiefel manifolds implemented in LDR toolbox [2][4]. They are stiefel, stiefel_trust, and stiefel_trust_prod for solving problem (5) (i.e., $\ell = 2$). stiefel and stiefel_trust are based on the alternating scheme as in Algorithm 1 except that all involved subproblems (13) are solved by the generic Riemannian steepest descent method and the Riemannian trust-region method [9], respectively. stiefel_trust_prod is the plain Riemannian Trust-Region (RTR) of [9] applied to (5) directly. We use the default settings of these three algorithms coded in the LDR toolbox, in which stiefel, stiefel_trust stop whenever the number of alternating steps $\nu > 100$ or

$$\left| \frac{f(X^{(\nu)}, Y^{(\nu)}) - f(X^{(\nu-1)}, Y^{(\nu-1)})}{f(X^{(\nu)}, Y^{(\nu)})} \right| \leq \epsilon_{\mathrm{alt}}, \qquad (33)$$

with $\epsilon_{\mathrm{alt}} = 10^{-8}$ and f is given by (4), whereas stiefel_trust_prod uses the default setting of RTR [34] in the package `manopt`. OCCA-scf is terminated if $\nu > 30$ or (33).

Our experiments are performed over synthetic data with $m = n = 1000$ for varying $k \in [3, 100]$, and the yeast data shown in Table 1 with $m = 101, n = 14$ for varying $k \in [2, 14]$. Following [2], we generate synthetic data with $q = 10^4$ and two views controlled by two sets of latent variables W and Z as follows:

4. https://github.com/cunni/ldr

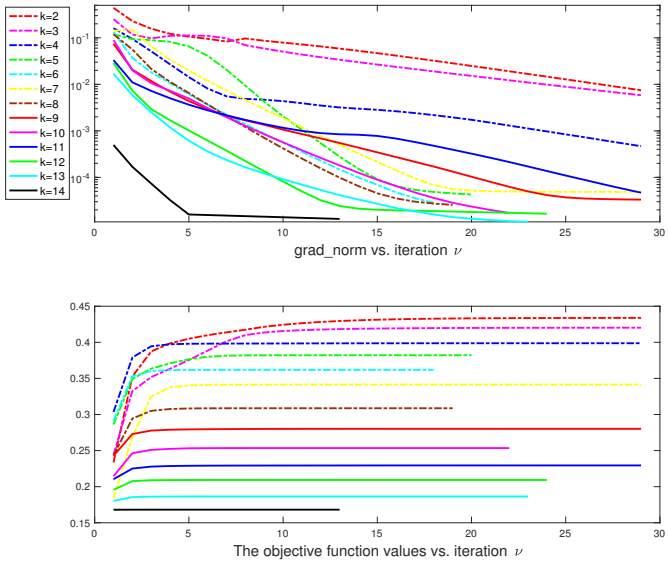


Fig. 2. Convergence curves of Algorithm 1 on the multi-label classification data yeast.

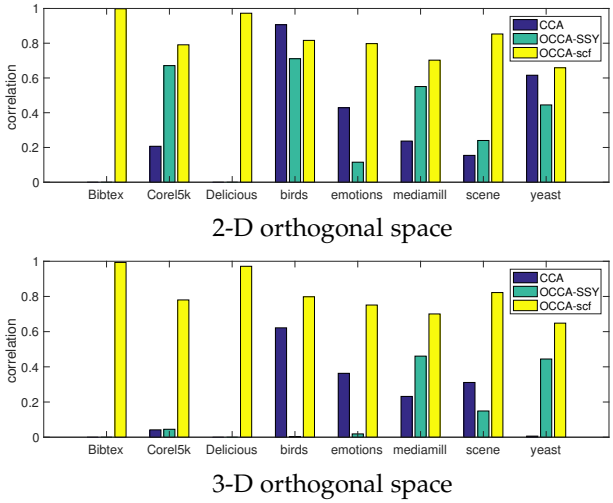


Fig. 3. Correlations obtained by three CCA methods in the 2-D and 3-D orthogonal spaces. The higher the bar is, the better the method performs.

$$d_z = \left\lceil \frac{\max(m,n)}{2} \right\rceil, d_w = \left\lceil \frac{2\max(m,n)}{5} \right\rceil,$$
$$S_X = P_X Z + Q_X W + \lambda E_X,\ S_Y = P_Y Z + Q_Y W + \lambda E_Y,$$

where $Z \in \mathbb{R}^{d_z \times q}$, $W \in \mathbb{R}^{d_w \times q}$, $P_X \in \mathbb{R}^{m \times d_z}$, $Q_X \in \mathbb{R}^{m \times d_w}$, $P_Y \in \mathbb{R}^{n \times d_z}$, $Q_Y \in \mathbb{R}^{n \times d_w}$, $E_X \in \mathbb{R}^{m \times q}$, and $E_Y \in \mathbb{R}^{n \times q}$ are matrices whose entries are i.i.d. sampled from a normal distribution with zero mean and unit standard deviation, and $\lambda = 2 \times 10^{-4}$.

The performance is evaluated in terms of the following three measurements:

1) Computational complexity measured by CPU time;
2) Correlation difference: they are computed by subtracting the objective value by the three stiefel methods from the one by OCCA-scf. The larger the difference is, the better OCCA-scf performs;
3) The 2-norm grad_norm of the Riemannian gradient (4) of f on the manifold $\mathbb{O}^{n\times k} \times \mathbb{O}^{m\times k}$ at an approximate solution.

Figure 1 shows the numerical results obtained by four different methods (all starting with the initial guess $(X^{(0)}, Y^{(0)}) = (I_{n,k}, I_{m,k})$ where $I_{n,k} \in \mathbb{R}^{n\times k}$ consists of the first k columns of I_n). We have the following observations:

1) For small k, OCCA-scf converges much faster than others. The CPU time by OCCA-scf is similar to stiefel, while stiefel_trust is most expensive among all.
2) OCCA-scf obtains similar correlation values on both datasets to stiefel_trust and stiefel_trust_prod. stiefel is worst. Correlation values of stiefel_trust_prod shows opposite trend as k increasing. This implies that stiefel is sensitive to k and input data.
3) Among the three stiefel-related methods, stiefel is fastest but is too inaccurate to be in the competition, and stiefel_trust achieves competitive accuracy but is too expensive to use. This leaves stiefel_trust_prod as the only one for further considerations.
4) Between OCCA-scf and stiefel_trust_prod, on the synthetic data, the former beats the latter, but on the yeast data the comparison is mixed: stiefel_trust_prod is slightly more accurate at an expense of about 10 times slower. Another advantage of OCCA-scf over stiefel_trust_prod is that the former can be easily extended to solve the multi-view problem, the OMCCA model (25), whose gradient over all of its arguments is rather messy and expensive to compute, making applying stiefel_trust_prod cumbersome.
5) OCCA-scf, stiefel, and stiefel_trust all adopt the same alternating scheme, except for their difference in how all involved subproblems (13) are solved. Yet, the latter two perform miserably. The major reason we think is the way how Algorithm 2 is designed: directly drive the Riemannian gradient to 0 while efficiently push up the objective.

So far as to n_{alt}, the number of full alternating iterations taken by OCCA-scf, we observed that the stopping criterion (33) is satisfied with $n_{\text{alt}} \leq 8$ on the synthetic data for all tested k. On the yeast data, however, for several $k \in [2, 14]$ the maximum number 30 of alternating iterations is reached, as shown in Figure 2. Despite that, the plots clearly show that the objective function has visually converged in these hard cases. In fact, Figure 1 demonstrates that the overall solution accuracy by OCCA-scf compares favorably with the generic optimization solvers.

6.3 Correlation Analysis and Data Visualization

In this section, we first explore the embedded subspaces obtained by three different CCA methods: the classical CCA, OCCA-SSY and OCCA-scf. For data visualization, the orthogonal spaces of 2-D and 3-D are the main focus. As aforementioned, the classical CCA method does not generate the orthonormal basis matrices for projection, and OCCA-SSY also does not guarantee to generate the orthonormal basis matrices either because of its numerical instability as mentioned before.

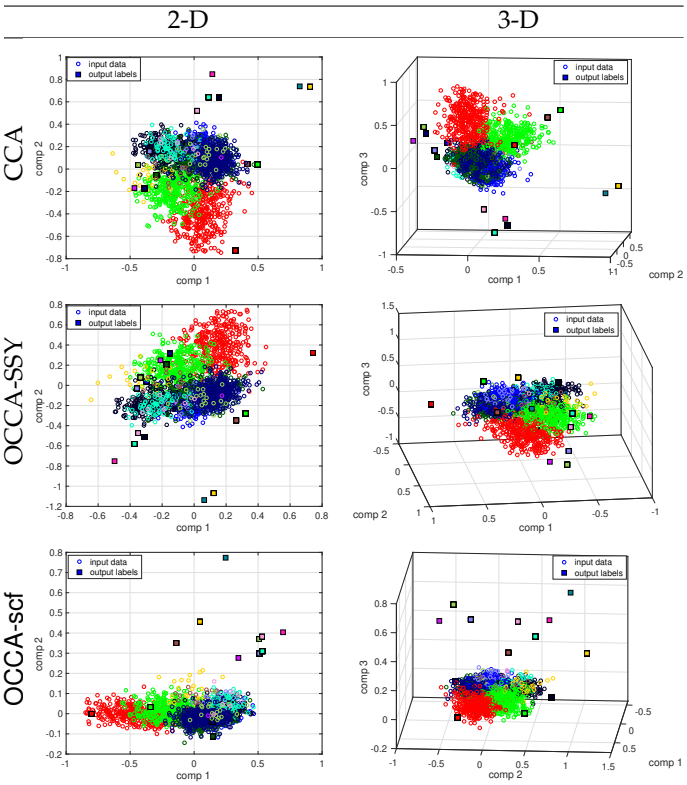


Fig. 4. Comparisons of three OCCA methods on the scene data in terms of 2-D and 3-D embeddings. Colors represent classes. The markers circle and square represent input data points and output classes. There are 15 classes extracted from 6 multiple labels.

We first investigate the quality of orthonormal basis matrices obtained by baseline methods in terms of correlation score. To obtain orthonormal basis matrices, we post-orthogonalize the columns of the basis matrices obtained by CCA and OCCA-SSY. (Note that the post-orthogonalization step is only applied in this experiment for studying the orthonormal property and data visualization.) If the matrices is rank deficient, we set the correlation to 0 since the number of orthogonal basis vectors is smaller than requested. Figure 3 shows the comparisons of three CCA methods in terms of the correlation score over eight real datasets in Table 1 for multi-label classification (detailed description is presented in Subsubsection 6.4.1). It can be seen that our proposed OCCA-scf achieves the best performance among all. More importantly, our method never encounters the matrix rank deficient issue, while it happens to CCA and OCCA-SSY on some datasets, such as Bibtex and Delicious.

We then explore the embeddings in 2-D and 3-D spaces and examine the correlations between the input data and its multi-label outputs. Since each sample may have multiple labels, we transform the multi-label classification problem into the multi-class classification problem using the label powerset approach [35] for the purpose of data visualization. The set of multiclass labels consists of all unique label combinations found in the data. For example, the data scene has 6 labels, and there are 15 unique label combinations in total. Figure 4 shows the embeddings of both input and output in 2-D and 3-D spaces colored by unique classes. Since multiple data points are assigned to the same unique class, there are only 15 embedded output points. In the cases of both 2-D and 3-D, our OCCA-scf method shows the best alignments between input data and output labels, for example, the majority of classes such as the red, green and blue ones are aligned best in the reduced space with the input data clouds.

TABLE 1
Datasets for multi-label classification

Dataset	Samples	Attributes	labels
birds	645	260	19
Corel5k	5000	499	374
emotions	593	72	6
scene	2407	294	6
yeast	2417	101	14
Bibtex	7395	1836	159
Mediamill	43903	120	101
Delicious	16105	500	983

6.4 Applications

We evaluate our proposed methods on two real-world applications for multi-label classification and multi-view feature extraction, where various CCA methods have been explored in the literature [4], [8], [36], [37], [38].

6.4.1 *Multi-label classification*

Multi-label classification [39] is a variant of the classification problem, where one instance may have various number of labels from a set of predefined categories, i.e., a subset of labels. It is different from multi-class classification, where each instance only has a single label. In general, the output class labels of one instance are represented by the indicator vector of size m where m is the number of class labels. If the cth label is assigned to the instance, the cth element of the indicator vector is 1, and otherwise 0. Let $S_1 \in \mathbb{R}^{n\times q}$ be the q instances of size n and $S_2 \in \mathbb{R}^{m\times q}$ be comprised of the q indicator vectors of size m. The popular use of CCA for multi-label classification is to treat X as one view and S_2 as the other view [36], [37], [38].

The multi-label classification datasets used in our experiments are the ones shown in Table 1. All datasets are publicly available[5]. Following [36], we take CCA as a supervised dimensionality reduction step for multi-label classification so that the embeddings obtained by CCA methods can encode certain correlations among input data and labels. Hence, it expects to have better performance for multi-label classification. Since some datasets have a small number of output labels, the reduced dimension is upper bounded by the number of output labels due to the inherent property of CCA. To alleviate the limitation from CCA and improve classification performance for general datasets, we propose to augment the learned embeddings using the original input data through a simple concatenation over two sets of features.

In this paper, we choose to use ML-kNN[6] as our backend multi-label classifier [40], which has demonstrated good performance over various datasets. We compare our OCCA-scf with other CCA methods including OCCA-SSY [4], LS-CCA [36] and classical CCA. All CCA-based methods take ML-kNN as the base classifier and corresponding augmentation approach for each CCA method is indicated by the

5. http://mulan.sourceforge.net/datasets-mlc.html
6. http://lamda.nju.edu.cn/files/MLkNN.rar

TABLE 2
Results in terms of the 5 measurements on the five datasets (40% for training and 60% for testing over 10 random splits). Best results are in bold.

dataset	method	HammingLoss	RankingLoss	OneError	Coverage	Average_Precision
birds	OCCA-scf	**0.0503 ± 0.0035**	**0.2173 ± 0.0062**	**0.4964 ± 0.0201**	**2.8866 ± 0.1580**	**0.5452 ± 0.0118**
	OCCA-scf-aug	0.0545 ± 0.0026	0.3045 ± 0.0047	0.7101 ± 0.0136	3.8597 ± 0.1754	0.3942 ± 0.0107
	CCA	0.1167 ± 0.0095	0.3509 ± 0.0197	0.8110 ± 0.0302	4.2028 ± 0.2954	0.3087 ± 0.0192
	CCA-aug	0.0545 ± 0.0026	0.3046 ± 0.0046	0.7101 ± 0.0136	3.8602 ± 0.1745	0.3942 ± 0.0107
	LS-CCA	0.1167 ± 0.0095	0.3509 ± 0.0197	0.8110 ± 0.0302	4.2028 ± 0.2954	0.3084 ± 0.0191
	LS-CCA-aug	0.0545 ± 0.0026	0.3046 ± 0.0046	0.7101 ± 0.0136	3.8602 ± 0.1745	0.3942 ± 0.0107
	OCCA-SSY	0.0618 ± 0.0049	0.2669 ± 0.0150	0.5978 ± 0.0269	3.4499 ± 0.2146	0.4722 ± 0.0182
	OCCA-SSY-aug	0.0545 ± 0.0026	0.3046 ± 0.0046	0.7101 ± 0.0136	3.8607 ± 0.1752	0.3942 ± 0.0108
	ML-kNN	0.0545 ± 0.0026	0.3046 ± 0.0046	0.7101 ± 0.0136	3.8607 ± 0.1752	0.3942 ± 0.0108
emotions	OCCA-scf	**0.2283 ± 0.0064**	**0.2016 ± 0.0091**	**0.3258 ± 0.0201**	**1.9643 ± 0.0456**	**0.7640 ± 0.0118**
	OCCA-scf-aug	0.2716 ± 0.0057	0.2799 ± 0.0098	0.3989 ± 0.0171	2.3862 ± 0.0573	0.6959 ± 0.0085
	CCA	0.2395 ± 0.0090	0.2204 ± 0.0138	0.3497 ± 0.0169	2.0736 ± 0.0730	0.7443 ± 0.0126
	CCA-aug	0.2718 ± 0.0059	0.2801 ± 0.0094	0.3986 ± 0.0168	2.3860 ± 0.0595	0.6960 ± 0.0082
	LS-CCA	0.2346 ± 0.0084	0.2088 ± 0.0149	0.3385 ± 0.0182	2.0096 ± 0.0930	0.7553 ± 0.0154
	LS-CCA-aug	0.2719 ± 0.0056	0.2795 ± 0.0099	0.3983 ± 0.0172	2.3848 ± 0.0572	0.6964 ± 0.0085
	OCCA-SSY	0.2577 ± 0.0141	0.2543 ± 0.0198	0.3860 ± 0.0274	2.2309 ± 0.0916	0.7190 ± 0.0172
	OCCA-SSY-aug	0.2719 ± 0.0057	0.2800 ± 0.0095	0.3986 ± 0.0170	2.3862 ± 0.0589	0.6958 ± 0.0084
	ML-kNN	0.2720 ± 0.0057	0.2798 ± 0.0097	0.3983 ± 0.0169	2.3862 ± 0.0589	0.6960 ± 0.0085
Scene	OCCA-scf	0.1214 ± 0.0024	0.1375 ± 0.0070	0.3329 ± 0.0073	0.7772 ± 0.0360	0.7902 ± 0.0060
	OCCA-scf-aug	**0.0941 ± 0.0016**	**0.0817 ± 0.0028**	**0.2428 ± 0.0081**	**0.4981 ± 0.0154**	0.8557 ± 0.0041
	CCA	0.1267 ± 0.0032	0.1448 ± 0.0068	0.3451 ± 0.0087	0.8153 ± 0.0369	0.7810 ± 0.0065
	CCA-aug	0.0949 ± 0.0020	0.0820 ± 0.0035	0.2440 ± 0.0080	0.4999 ± 0.0194	0.8555 ± 0.0044
	LS-CCA	0.1228 ± 0.0028	0.1401 ± 0.0059	0.3361 ± 0.0085	0.7909 ± 0.0326	0.7873 ± 0.0058
	LS-CCA-aug	0.0948 ± 0.0021	0.0821 ± 0.0034	0.2440 ± 0.0082	0.5003 ± 0.0187	0.8553 ± 0.0044
	OCCA-SSY	0.1183 ± 0.0030	0.1302 ± 0.0055	0.3226 ± 0.0086	0.7405 ± 0.0314	0.7979 ± 0.0063
	OCCA-SSY-aug	0.0943 ± 0.0020	0.0818 ± 0.0025	0.2431 ± 0.0090	0.4981 ± 0.0137	**0.8558 ± 0.0042**
	ML-kNN	0.0949 ± 0.0020	0.0823 ± 0.0033	0.2442 ± 0.0085	0.5009 ± 0.0188	0.8554 ± 0.0042
Corel5k	OCCA-scf	**0.0094 ± 0.0000**	**0.1365 ± 0.0015**	**0.7252 ± 0.0055**	**116.3179 ± 1.2727**	**0.2529 ± 0.0034**
	OCCA-scf-aug	**0.0094 ± 0.0000**	0.1373 ± 0.0016	0.7309 ± 0.0067	116.9343 ± 1.3411	0.2474 ± 0.0031
	CCA	**0.0094 ± 0.0000**	0.1396 ± 0.0012	0.7519 ± 0.0079	117.9406 ± 1.1541	0.2339 ± 0.0035
	CCA-aug	**0.0094 ± 0.0000**	0.1381 ± 0.0016	0.7327 ± 0.0056	117.3671 ± 1.2944	0.2436 ± 0.0031
	LS-CCA	0.0095 ± 0.0000	0.1379 ± 0.0015	0.7432 ± 0.0082	116.8526 ± 1.3332	0.2463 ± 0.0027
	LS-CCA-aug	**0.0094 ± 0.0000**	0.1376 ± 0.0015	0.7323 ± 0.0082	117.1294 ± 1.2682	0.2459 ± 0.0039
	OCCA-SSY	**0.0094 ± 0.0000**	**0.1365 ± 0.0015**	0.7263 ± 0.0085	116.4133 ± 1.3187	0.2522 ± 0.0038
	OCCA-SSY-aug	**0.0094 ± 0.0000**	0.1371 ± 0.0016	0.7304 ± 0.0054	116.8367 ± 1.2580	0.2481 ± 0.0032
	ML-kNN	**0.0094 ± 0.0000**	0.1381 ± 0.0019	0.7323 ± 0.0062	117.5434 ± 1.4205	0.2434 ± 0.0035
yeast	OCCA-scf	0.2080 ± 0.0021	0.1838 ± 0.0039	0.2538 ± 0.0066	6.5615 ± 0.0736	0.7445 ± 0.0049
	OCCA-scf-aug	**0.1997 ± 0.0033**	**0.1735 ± 0.0034**	**0.2356 ± 0.0075**	**6.3870 ± 0.0859**	**0.7556 ± 0.0041**
	CCA	0.2108 ± 0.0031	0.1894 ± 0.0056	0.2539 ± 0.0069	6.6438 ± 0.0969	0.7364 ± 0.0054
	CCA-aug	0.2011 ± 0.0026	0.1762 ± 0.0035	0.2398 ± 0.0068	6.4152 ± 0.0827	0.7512 ± 0.0040
	LS-CCA	0.2077 ± 0.0038	0.1855 ± 0.0044	0.2518 ± 0.0092	6.5928 ± 0.1040	0.7436 ± 0.0054
	LS-CCA-aug	0.2012 ± 0.0028	0.1760 ± 0.0036	0.2405 ± 0.0060	6.4096 ± 0.0864	0.7511 ± 0.0038
	OCCA-SSY	0.2097 ± 0.0035	0.1871 ± 0.0052	0.2526 ± 0.0058	6.6265 ± 0.1023	0.7403 ± 0.0055
	OCCA-SSY-aug	0.1997 ± 0.0033	0.1740 ± 0.0033	0.2356 ± 0.0074	6.3907 ± 0.0984	0.7554 ± 0.0036
	ML-kNN	0.2017 ± 0.0029	0.1759 ± 0.0036	0.2397 ± 0.0067	6.4075 ± 0.0831	0.7512 ± 0.0036

TABLE 3
Multi-view datasets

Dataset	Samples	Multiple views	classes
mfeat	2000	216;76;64;6;240;47	10
Caltech101-7	1474	254;512;1180;1008;64;1000	7
Caltech101-20	2386	254;512;1180;1008;64;1000	20
Scene15	4310	254;512;531;360;64;1000	15
yeast_ribosomal	1040	3735;4901;441	2

suffix "-aug". We randomly split the data into 40% for training and 60% for testing and tune the neighborhood parameter within the set $\{1, 3, 5, 7, 9, 13, 15\}$ for ML-kNN. Following [40], we report the best results and their standard deviations over 10 random train/test splits in terms of the following five measurements:

- HammingLoss: the average number of times an instance-label pair is misclassified.
- RankingLoss: the average fraction of label pairs that are reversely ordered for the instance.
- OneError: the average number of times the top-ranked label is not in the set of proper labels of the instance
- Coverage: on the average how far we need to go down the list of labels in order to cover all the proper labels of the instance.
- Average_Precision: the average precision of labels ranked above a particular label in the same label set.

Except Average_Precision, the first four measurements show good performances of multi-label classification if the measurement value is small.

Table 2 shows the results obtained by the compared methods over the five datasets in terms of the five measurements. We observe that our OCCA-scf and OCCA-scf-aug show the best results on almost all the five measurements except Average_Precision on Scene by OCCA-SSY-aug. For datasets scene and yeast, OCCA-scf-aug shows better results than OCCA-scf. This implies that our augmentation approach is effective when the features obtained by dimensionality reduction method such as CCA somehow lose the

TABLE 4
Means and standard deviations of accuracy obtained by 1-nearest neighbor classifier on each view and embeddings obtained by three CCA methods over 10 random draws from each dataset (30% training and 70% testing). Parameter k used by CCA methods to achieve the best accuracy is shown in the bracket. The symbol "-" is for the non-existence of the view.

	mfeat	Caltech101-7	Caltech101-20	Scene15	yeast_ribosomal
view1	0.9513 ± 0.0053	0.9259 ± 0.0049	0.7659 ± 0.0046	0.5766 ± 0.0091	0.8553 ± 0.0472
view2	0.7604 ± 0.0104	0.9443 ± 0.0051	0.8257 ± 0.0064	0.5269 ± 0.0070	0.8831 ± 0.0072
view3	0.9293 ± 0.0043	0.9415 ± 0.0070	0.8226 ± 0.0106	0.5528 ± 0.0081	**0.9856 ± 0.0046**
view4	0.6780 ± 0.0064	0.9287 ± 0.0105	0.7968 ± 0.0118	0.4609 ± 0.0079	-
view5	0.9630 ± 0.0025	0.7759 ± 0.0133	0.6042 ± 0.0122	0.6946 ± 0.0130	-
view6	0.7814 ± 0.0077	0.9152 ± 0.0059	0.7645 ± 0.0128	-	-
MCCA	0.8679 ± 0.0073 (6)	0.8865 ± 0.0072 (15)	0.8620 ± 0.0072 (40)	0.6851 ± 0.0043 (35)	0.8155 ± 0.0139 (3)
OMCCA-SS	0.8298 ± 0.0089 (6)	0.9493 ± 0.0024 (45)	0.8527 ± 0.0057 (50)	0.7030 ± 0.0081 (50)	0.8379 ± 0.0110 (5)
RCOMCCA-G (uniform)	0.7634 ± 0.0134 (5)	0.8880 ± 0.0052 (50)	0.7150 ± 0.0075 (45)	0.4866 ± 0.0044 (50)	0.8639 ± 0.0291 (40)
RCOMCCA-G (top-p)	**0.9696 ± 0.0035** (5)	**0.9664 ± 0.0060** (35)	0.8887 ± 0.0077 (25)	0.7542 ± 0.0054 (30)	0.8756 ± 0.0095 (45)
RCOMCCA-G (tree)	0.9566 ± 0.0031 (6)	0.9392 ± 0.0043 (45)	0.7882 ± 0.0078 (50)	0.4004 ± 0.0063 (30)	0.8678 ± 0.0161 (45)
RCOMCCA-J (uniform)	0.7540 ± 0.0121 (5)	0.8868 ± 0.0068 (30)	0.7350 ± 0.0091 (50)	0.4995 ± 0.0059 (50)	0.8492 ± 0.0201 (35)
RCOMCCA-J (top-p)	0.9692 ± 0.0038 (5)	0.9649 ± 0.0029 (15)	**0.8893 ± 0.0074** (25)	**0.7574 ± 0.0077** (30)	0.8782 ± 0.0071 (35)
RCOMCCA-J (tree)	0.9581 ± 0.0055 (6)	0.9474 ± 0.0041 (45)	0.7799 ± 0.0084 (50)	0.4188 ± 0.0123 (35)	0.8678 ± 0.0099 (25)

information that is also useful for multi-label classification although the correlations remain. It is worth noting that our methods outperform ML-kNN over all experimented datasets. These observations imply that OCCA with orthogonality constraints improves ML-kNN for multi-label classification and our proposed OCCA-scf methods outperform other CCA methods.

6.4.2 Multi-view feature extraction

Previous experiments focus on problems with only two views. Here, we aim to evaluate our proposed RCOMCCA in terms of multi-view feature extraction [4], [8]. Following [4], we employ the serial fusion strategy to concatenate embeddings from all views for classification based on 1-nearest neighbor classifier. Since LDR-based CCA and LS-CCA are not easy to be extended for learning with multiple views, we compare our proposed RCOMCCA with MCCA [14], [15] and OMCCA-SS [8]. For the top-p weighting scheme, $p \in \{1, 3, 6\}$ is used, except that $p \in \{1, 3\}$ is used for dataset yeast_ribosomal.

The datasets with relevant statistics are shown in Table 3. For image datasets such as Caltech101[7] [41] and Scene15[8] [42], we apply various feature descriptors to extract features of views including CENTRIST [43], GIST [44], LBP [45], histogram of oriented gradient (HOG), color histogram (CH), and SIFT-SPM [42]. Note that we drop CH for Scene15 due to the gray-level images. mfeat is the handwritten numeral data[9] [46] with 6 views including 76 Fourier coefficients of the character shapes, 216 profile correlations, 64 Karhunen-Love coefficients, 240 pixel averages in 2×3 windows, 47 Zernike moments, and 6 morphological features. The Berkeley genomic dataset yeast_ribosomal[10] is used where three aspects of the protein are considered as the views including Pfam HMM, Hydrophobicity FFT and Gene expression for binary classification, e.g., ribosomal vs. non-ribosomal.

We use 1-nearest neighbor classifier as the base classifier for evaluating the performance of multi-view feature extraction. We run CCA methods to generate embeddings by varying $k \in \{3, 4, 5, 6\}$ for mfeat, and $k \in \{3, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ for other datasets. We split the data into training and testing with the ratio 30/70. Classification accuracy is used as the performance evaluation criterion. Experimental results are reported in terms of the average of 10 randomly drawn splits.

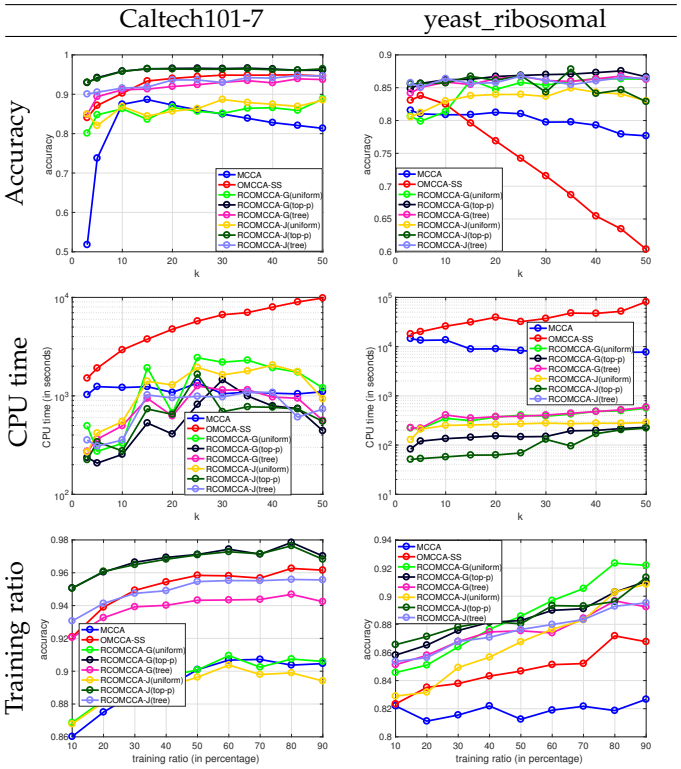


Fig. 5. Accuracy and CPU time of three MCCA methods on four datasets for varying the reduced dimension k and the training ratio.

We first compare eight variants of CCA methods and the classifiers based on each single view using all input features. Table 4 shows the results over five multi-view datasets with the best k shown in bracket for each method. From Table 4, we have the following observations:

1) CCA-based methods can achieve competitive or better results using a small set of features comparing with the best single view of the input features.
2) OCCA methods including RCOMCCA (top-p) and OMCCA-SS generally show better results than classi-

7. http://www.vision.caltech.edu/Image_Datasets/Caltech101/
8. https://figshare.com/articles/15-Scene_Image_Dataset/7007177
9. https://archive.ics.uci.edu/ml/datasets/Multiple+Features
10. https://noble.gs.washington.edu/proj/sdp-svm/

cal MCCA. This implies that orthogonality constraints added to MCCA can improve learning performance.

3) Our proposed RCOMCCA methods with the top-p weighting scheme demonstrates much better results than MCCA and OMCCA-SS can by large margins. Except for yeast_ribosomal, RCOMCCA-G (top-p) and RCOMCCA-J (top-p) outperform the classifier of the best single view on the other four datasets.
4) RCOMCCA with the top-p weighting scheme outperforms RCOMCCA with other two weighting schemes. This implies that pairs of views can contribute differently to the downstreaming classification problem.
5) For the same weighting schemes, our proposed RCOMCCA methods with Gauss-Seidel-style and Jacobi-style yield almost similar results. It is recommended to take the problem structure into account for selecting the proper solver for efficiency as discussed in Section 5.

We also compare eight variants of CCA methods in terms of three other measurements including the sensitivity of parameter k, CPU time, and sampling ratio of training and testing data. The results are shown in Figure 5. It is clear to see that

1) Accuracies of all CCA methods increase with k. However, MCCA on Caltech101-7 and OMCCA-SS on yeast_ribosomal behave abnormally since their performances degrade significantly after a few small k.
2) RCOMCCA generally is the most efficient method among the three methods. Due to its incremental optimization scheme, OMCCA-SS takes linear computational complexity with k, and so its CPU time increases with k. MCCA becomes less efficient if the total number of features in all views are large, for example yeast_ribosomal, because it has to solve the generalized eigenvalue problem whose size is the sum of the numbers of features in all views. As shown in Figure 5, MCCA on yeast_ribosomal takes more than 10 times longer than RCOMCCA.
3) All methods show better performances when the number of training data increases. One notable exception is MCCA on yeast_ribosomal, which does not show much gain as training data ratio increases significantly. All orthogonally constrained CCA methods do not show this issue.

These observations demonstrate that our proposed RCOMCCA not only can achieve noticeably better performance but also is much faster than OMCCA-SS and MCCA for multi-view feature extraction.

7 Conclusions

In this paper, we start by proposing an efficient way for solving CCA with orthogonality constraints, called the orthogonal CCA (OCCA). Then to model the data with more than two views, we present a novel weighted multiset CCA again with orthogonality constraints (OMCCA). Our algorithms rely on the solution of a subproblem with trace-fractional structure, which is solved by a newly proposed SCF iteration. Theoretically, we perform a global and local convergence analysis. Extensive experiments are conducted to evaluate the proposed algorithms against existing methods in terms of various measurements, such as parameter sensitivity, correlation, computational time, and data visualization. We further apply our methods to real-world applications for multi-label classification and multi-view feature extraction. Experimental results show that our methods not only perform competitively to or significantly better than baselines in terms of accuracy but also are more efficient. This work focuses on the linear orthogonal projection. In the future, we would like to explore similar ideas for nonlinear CCA and other variants of CCA methods with orthogonality constraints.

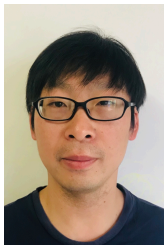
Acknowledgements

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

References

[1] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3-4, pp. 321–377, 1936.
[2] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. Mach. Learning Res.*, vol. 16, pp. 2859–2900, 2015.
[3] 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, p. 95, 2018.
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms On Matrix Manifolds*. Princeton University Press, 2008.
[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, Maryland: Johns Hopkins University Press, 2013.
[11] 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.
[12] L.-H. Zhang and R.-C. Li, "Maximization of the sum of the trace ratio on the Stiefel manifold, I: Theory," *SCIENCE CHINA Math.*, vol. 57, no. 12, pp. 2495–2508, 2014.
[13] ——, "Maximization of the sum of the trace ratio on the Stiefel manifold, II: Computation," *SCIENCE CHINA Math.*, vol. 58, no. 7, pp. 1549–1566, 2015.
[14] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.
[15] 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.
[16] 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.
[17] B. N. Parlett, *The Symmetric Eigenvalue Problem*. Philadelphia: SIAM, 1998.

[18] Y. Cai, L.-H. Zhang, Z. Bai, and R.-C. Li, "On an eigenvector-dependent nonlinear eigenvalue problem," *SIAM J. Matrix Anal. Appl.*, vol. 39, no. 3, pp. 1360–1382, 2018.
[19] L.-H. Zhang, L.-Z. Liao, and M. K. Ng, "Fast algorithms for the generalized Foley-Sammon discriminant analysis," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 4, pp. 1584–1605, 2010.
[20] L. Li and Z. Zhang, "Semi-supervised domain adaptation by covariance matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, to appear.
[21] L.-H. Zhang, "Uncorrelated trace ratio LDA for undersampled problems," *Patt. Recog. Lett.*, vol. 32, pp. 476–484, 2011.
[22] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*. Cambridge, UK: Cambridge University Press, 2004.
[23] Y. Saad, J. R. Chelikowsky, and S. M. Shontz, "Numerical methods for electronic structure calculations of materials," *SIAM Rev.*, vol. 52, no. 1, pp. 3–54, 2010.
[24] Z. Bai, D. Lu, and B. Vandereycken, "Robust rayleigh quotient minimization and nonlinear eigenvalue problems," *SIAM J. Sci. Comput.*, vol. 40, pp. A3495–A3522, 2018.
[25] Z. Wang, Q. Ruan, and G. An, "Projection-optimal local Fisher discriminant analysis for feature extraction," *Neural Comput & Applic*, vol. 26, pp. 589–601, 2015.
[26] R.-C. Li, "Rayleigh quotient based optimization methods for eigenvalue problems," in *Matrix Functions and Matrix Equations*, ser. Series in Contemporary Applied Mathematics, Z. Bai, W. Gao, and Y. Su, Eds. Singapore: World Scientific, 2015, vol. 19, pp. 76–108, lecture summary for 2013 Gene Golub SIAM Summer School.
[27] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*. Boston: Academic Press, 1990.
[28] J.-G. Sun, *Matrix Perturbation Analysis*. Beijing: Academic Press, 1987, in Chinese.
[29] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge: Cambridge University Press, 1991.
[30] J. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA: SIAM, 1997.
[31] E. Anderson, Z. Bai, C. Bischof, J. W. 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.
[32] A. V. Knyazev, "Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method," *SIAM J. Sci. Comput.*, vol. 23, no. 2, pp. 517–541, 2001.
[33] A. V. Knyazev and K. Neymeyr, "Efficient solution of symmetric eigenvalue problems using multigrid preconditioners in the locally optimal block conjugate gradient method," *Electron. Trans. Numer. Anal.*, vol. 15, pp. 38–55, 2003.
[34] P.-A. Absil, C. G. Baker, and K. A. Gallivan, "Trust-region methods on Riemannian manifolds," *Found. Comput. Math.*, vol. 7, no. 3, pp. 303–330, 2007.
[35] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.
[36] L. Sun, S. Ji, and J. Ye, "Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 194–200, 2010.
[37] Y. Zhang and J. Schneider, "Multi-label output codes using canonical correlation analysis," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 873–882.
[38] P. Rai and H. Daume, "Multi-label prediction via sparse infinite cca," in *Advances in Neural Information Processing Systems*, 2009, pp. 1518–1526.
[39] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.
[40] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.
[41] 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.
[42] 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.
[43] 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.
[44] 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.
[45] 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.
[46] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml



Lei-Hong Zhang is a professor with the School of Mathematical Sciences, Soochow University, Suzhou, 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 numerical optimization, eigenvalue problems and machine learning.

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.

Zhaojun Bai is a professor in the Department of Computer Science and Department of Mathematics, University of California, Davis, CA, USA. He obtained his PhD from Fudan University, China and postdoctoral fellowship from Courant Institute, New York University. His main research interests include linear algebra algorithm design and analysis, mathematical software engineering and applications in computational science and engineering and data science. He participated a number of synergistic projects, such as LAPACK. He is an Editor-in-Chief of ACM TOMS, and serves on editorial boards of JCM and Science China Mathematics among others. Previously, he served as an associate editor of SIMAX, vice chair of IEEE IPDPS and numerous other professional positions. He is a Fellow of SIAM.

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.