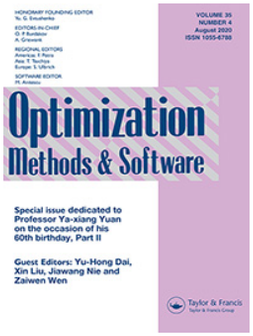

Optimization Methods and Software

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/goms20

Orthogonal canonical correlation analysis and applications

Li Wang , Lei-hong Zhang , Zhaojun Bai & Ren-Cang Li

To cite this article: Li Wang , Lei-hong Zhang , Zhaojun Bai & Ren-Cang Li (2020) Orthogonal canonical correlation analysis and applications, Optimization Methods and Software, 35:4, 787-807, DOI: 10.1080/10556788.2019.1700257

To link to this article: https://doi.org/10.1080/10556788.2019.1700257

Published online: 20 Jan 2020.

OPTIMIZATION METHODS & SOFTWARE
2020, VOL. 35, NO. 4, 787–807
https://doi.org/10.1080/10556788.2019.1700257

Orthogonal canonical correlation analysis and applications*

Li Wang[a], Lei-hong Zhang[b], Zhaojun Bai[c] and Ren-Cang Li[a]

[a]Department of Mathematics, University of Texas at Arlington, Arlington, TX, USA; [b]School of Mathematical Sciences, Soochow University, Suzhou, Jiangsu, People's Republif of China; [c]Department of Computer Science and Department of Mathematics, University of California, Davis, CA, USA

ABSTRACT

Canonical correlation analysis (CCA) is a cornerstone of linear dimensionality reduction techniques that jointly maps two datasets to achieve maximal correlation. CCA has been widely used in applications for capturing data features of interest. In this paper, we establish a range constrained orthogonal CCA (OCCA) model and its variant and apply them for three data analysis tasks of datasets in real-life applications, namely unsupervised feature fusion, multi-target regression and multi-label classification. Numerical experiments show that the OCCA and its variant produce superior accuracy compared to the traditional CCA.

ARTICLE HISTORY
Received 27 May 2019
Accepted 29 November 2019

KEYWORDS
Canonical correlation analysis (CCA); orthogonal CCA; singular value decomposition; unsupervised feature fusion; multi-target regression; multi-label classification

AMS SUBJECT CLASSIFICATIONS
15A18; 15A21; 62H20; 62H25; 65F15; 65F30

1. Introduction

Originally proposed by Hotelling in 1936 [16], canonical correlation analysis (CCA) is a classical linear dimensionality reduction technique that jointly maps two datasets to achieve maximal correlation. Modern treatments and enrichments include [13,21]. Specifically, given two datasets in the form of two data matrices

$$X_a \in \mathbb{R}^{n\times q}, \quad X_b \in \mathbb{R}^{m\times q}, \tag{1}$$

respectively, where n and m are the dimensions of the two data sets, respectively, and q is the number of data points in each of the two sets. Without loss of generality, we may assume that both X_a and X_b are centred, i.e. $X_a\mathbf{1}_q = 0$ and $X_b\mathbf{1}_q = 0$, where $\mathbf{1}_q \in \mathbb{R}^q$ is the vector of all ones; otherwise, we may preprocess X_a and X_b as

$$X_a \leftarrow X_a - \frac{1}{q}(X_a\mathbf{1}_q)\mathbf{1}_q^{\mathrm{T}}, \quad X_b \leftarrow X_b - \frac{1}{q}(X_b\mathbf{1}_q)\mathbf{1}_q^{\mathrm{T}}.$$

Sometimes the columns of X_a and X_b are normalized, too, to unit vectors, although not always. For a given pair of transformation vectors $\{p_a, p_b\}$, the canonical correlation

CONTACT Zhaojun Bai zbai@ucdavis.edu Department of Computer Science and Department of Mathematics, University of California, Davis, CA 95616, USA
*Verion date November 28, 2019.
Dedicated to Yaxiang Yuan on the occasion of his 60th birthday

$\rho(p_a, p_b)$ between $p_a^{\mathrm{T}} X_a$ and $p_b^{\mathrm{T}} X_b$ is given by

$$\rho(p_a, p_b) = \frac{p_a^{\mathrm{T}} C p_b}{\sqrt{(p_a^{\mathrm{T}} A p_a)(p_b^{\mathrm{T}} B p_b)}},$$

where

$$A = X_a X_a^{\mathrm{T}} \in \mathbb{R}^{n\times n}, \quad B = X_b X_b^{\mathrm{T}} \in \mathbb{R}^{m\times m}, \quad C = X_a X_b^{\mathrm{T}} \in \mathbb{R}^{n\times m}. \tag{2}$$

The traditional CCA seeks first pair $\{p_{a1}, p_{b1}\}$ of canonical vectors by maximizing $\rho(p_a, p_b)$ under the constraint $p_{a1}^{\mathrm{T}} A p_{a1} = p_{b1}^{\mathrm{T}} B p_{b1} = 1$ (see Remark 2.1 below for more discussions on constraints). For a positive integer k ($1 \le k \le \min\{m, n, q\}$), the pairs (p_{ai}, p_{bi}) for $i = 2, 3, \ldots, k$ then can be obtained sequentially by maximizing canonical correlation $\rho(p_a, p_b)$ subject to additional A- and B-orthogonality constraints for p_a and p_b, respectively, against those pairs that are already computed. It turns out (see, e.g. [5]) that the transformation matrices $P_a = [p_{a1}, \ldots, p_{ak}]$, $P_b = [p_{b1}, \ldots, p_{bk}]$ of the traditional CCA can be equivalently obtained by the following optimization problem

$$\max_{P_a\in\mathbb{R}^{n\times k},\, P_b\in\mathbb{R}^{m\times k}} \quad \frac{\mathrm{tr}(P_a^{\mathrm{T}} C P_b)}{\sqrt{\mathrm{tr}(P_a^{\mathrm{T}} A P_a)\mathrm{tr}(P_b^{\mathrm{T}} B P_b)}}, \tag{3a}$$

$$\text{subject to} \quad P_a^{\mathrm{T}} A P_a = P_b^{\mathrm{T}} B P_b = I_k. \tag{3b}$$

The traditional CCA (3) is well-posed only when $k \le \min\{\mathrm{rank}(X_a), \mathrm{rank}(X_b)\}$; otherwise, for example, if $k > \mathrm{rank}(X_a) = \mathrm{rank}(A)$, then $P_a^{\mathrm{T}} A P_a = I_k$ can never be satisfied. The solution of (3) is not unique. In fact, if (P_a, P_b) is a pair of the maximizers and so is $(-P_a, -P_b)$. Such non-uniqueness bas been conveniently overlooked. The solution of (3) can be obtained by solving a generalized eigenvalue problem or the singular value decomposition (SVD) and some efficient numerical techniques can be found in, e.g. [3,13,23] (see also Section 2.1). In terms of the SVD, a complete description of the solutions to (3) can be found in [3, Theorem 3.2].

The optimal P_a and P_b of CCA (3) in general do not have orthonormal columns and that can be disadvantageous and less effective, as argued in [5]. Orthogonal CCA (OCCA) is a term that was coined broadly as a collection of variants of the traditional CCA (3) that produce two matrices P_a and P_b with orthonormal columns to serve practical purposes similar to those by the ones of the traditional CCA.

Our main goals in this paper are two-fold: (1) establish a range constrained OCCA model and a variant and (2) apply them to three data science tasks, namely unsupervised feature fusion, multi-target regression, and multi-label classification. Our model is inspired by that the pair (P_a, P_b) of maximizers to CCA (3) can be constructed one column of each P_a and P_b at a time [3, Theorem 3.1]. A similar idea was used by Shen *et al.* [24], but we introduce range constraints that enable us to design algorithms based on the SVD, a widely used and well-proven numerical linear algebra technique. For this reason, our algorithms are robust, whereas the algorithm in [24] may not work properly. This will be discussed in detail in Section 2.2. Our models, together with our numerical solutions, often improve the results by CCA. Sometimes the improvements are dramatic.

The rest of this paper is organized as follows. Section 2.1 introduces range constrained CCA which will serve as the building block for implementing new CCA variants in the

rest of the paper. In Section 2.2, we start by stating the orthogonal CCA (OCCA) model of Shen *et al.* [24] and explaining its major numerical difficulty, and then propose a range constrained OCCA model and its robust implementation based on the well-proven SVD. A partial OCCA model is discussed in Section 4. Section 5 details our extensive numerical experiments that aim to demonstrate superior performance of our OCCA models on real-world data for three data science tasks. Finally, conclusions and remarks are drawn in Section 6.

Notation: $\mathbb{R}^{m\times n}$ is the set of all $m \times n$ real matrices, $\mathbb{R}^n = \mathbb{R}^{n\times 1}$, and $\mathbb{R} = \mathbb{R}^1$. I_n (or simply I if its dimension is clear from the context) is the $n \times n$ identity matrix. The superscript '$\cdot^{\mathrm{T}}$' takes transpose. For a matrix X, $\mathcal{R}(X)$ is the column space. For a vector $x \in \mathbb{R}^n$, $\|x\|_2 = \sqrt{x^{\mathrm{T}}x}$ is its ℓ_2-norm, and $\|x\|_B = \sqrt{x^{\mathrm{T}}Bx}$ is its B-semi-norm, where $B \in \mathbb{R}^{n\times n}$ is positive semi-definite. Throughout the rest of this paper, X_a, X_b, A, B, C are reserved as the ones given in (1) and (2) except within our algorithm descriptions, where X_a and X_b are updated. But we do not think this will cause any confusion.

2. Related work

2.1. Range constrained CCA

Intuitively, the maximization in (3) looks for the transformation matrix pair (P_a, P_b) such that $X_a^{\mathrm{T}}P_a$ and $X_b^{\mathrm{T}}P_b$ are best aligned. Any component of P_a and that of P_b in the orthogonal complements of $\mathcal{R}(X_a)$ and $\mathcal{R}(X_b)$ are annihilated in the calculations of $X_a^{\mathrm{T}}P_a$ and $X_b^{\mathrm{T}}P_b$, respectively. Thus it is sensible to also enforce, besides (3b), the range constraints $\mathcal{R}(P_a) \subset \mathcal{R}(X_a)$ and $\mathcal{R}(P_b) \subset \mathcal{R}(X_b)$. As a result, we are naturally led to a variant of the traditional CCA (3), the so-called *range constrained CCA*:

$$\max_{P_a\in\mathbb{R}^{n\times k},\, P_b\in\mathbb{R}^{m\times k}} \quad \frac{\mathrm{tr}(P_a^{\mathrm{T}}CP_b)}{\sqrt{\mathrm{tr}(P_a^{\mathrm{T}}AP_a)\mathrm{tr}(P_b^{\mathrm{T}}BP_b)}}, \tag{4a}$$

$$\text{subject to} \quad P_a^{\mathrm{T}}AP_a = P_b^{\mathrm{T}}BP_b = I_k, \tag{4b}$$

$$\mathcal{R}(P_a) \subset \mathcal{R}(X_a),\ \mathcal{R}(P_b) \subset \mathcal{R}(X_b). \tag{4c}$$

This variant can be solved by the SVD as follows (see also [3]). Let the SVDs of X_a and X_b be

$$X_a = U_a\Sigma_a V_a^{\mathrm{T}}, \quad U_a \in \mathbb{R}^{n\times r_a}, \quad V_a \in \mathbb{R}^{q\times r_a}, \quad \Sigma_a \in \mathbb{R}^{r_a\times r_a}, \tag{5a}$$

$$X_b = U_b\Sigma_b V_b^{\mathrm{T}}, \quad U_b \in \mathbb{R}^{m\times r_b}, \quad V_b \in \mathbb{R}^{q\times r_b}, \quad \Sigma_b \in \mathbb{R}^{r_b\times r_b}, \tag{5b}$$

where $r_a = \mathrm{rank}(X_a)$ and $r_b = \mathrm{rank}(X_b)$, and, without loss of generality, the diagonal entries of Σ_a and Σ_b are arranged in nonincreasing order. With the SVDs in (5), we have for A, B, and C defined in (2)

$$A = U_a\Sigma_a^2U_a^{\mathrm{T}}, \quad B = U_b\Sigma_b^2U_b^{\mathrm{T}}, \quad C = U_a\Sigma_aV_a^{\mathrm{T}}V_b\Sigma_bU_b^{\mathrm{T}}.$$

Set $\widehat{P}_a = \Sigma_aU_a^{\mathrm{T}}P_a \in \mathbb{R}^{r_a\times k}$ and $\widehat{P}_b = \Sigma_bU_b^{\mathrm{T}}P_b \in \mathbb{R}^{r_b\times k}$. Under (4c), we will have

$$P_a = U_a\Sigma_a^{-1}\widehat{P}_a, \quad P_b = U_b\Sigma_b^{-1}\widehat{P}_b. \tag{6}$$

The optimization problem (4) is then transformed into an equivalent problem:

$$\max_{\widehat{P}_a^{\mathrm{T}}\widehat{P}_a=\widehat{P}_b^{\mathrm{T}}\widehat{P}_b=I_k} \operatorname{tr}(\widehat{P}_a^{\mathrm{T}} V_a^{\mathrm{T}} V_b \widehat{P}_b), \tag{7}$$

which can also be solved by the SVD. Specifically, let

$$V_a^{\mathrm{T}} V_b = U \Sigma V^{\mathrm{T}} \tag{8}$$

be the SVD of $V_a^{\mathrm{T}} V_b$, where $U \in \mathbb{R}^{n\times r}$, $V \in \mathbb{R}^{m\times r}$, $r = \operatorname{rank}(V_a V_b^{\mathrm{T}})$, and $\Sigma \in \mathbb{R}^{r\times r}$ is diagonal. Then a pair of maximizers for (7) can be given by

$$(\widehat{P}_a, \widehat{P}_b) = \begin{cases} (U_{(:,1:k)}, V_{(:,1:k)}) & \text{if } r \geq k, \\ ([U, U_\perp], [V, V_\perp]) & \text{if } r < k, \end{cases} \tag{9}$$

where, in the case of $r < k$, $U_\perp \in \mathbb{R}^{n\times(r-k)}$ and $V_\perp \in \mathbb{R}^{m\times(r-k)}$ can be arbitrary so long as $\widehat{P}_a^{\mathrm{T}}\widehat{P}_a = \widehat{P}_b^{\mathrm{T}}\widehat{P}_b = I_k$ are ensured. Finally, a pair of maximizers P_a and P_b for the range constrained CCA (4) can be recovered by (6) with (9).

2.2. Orthogonal CCA

Although, solutions to the traditional CCA (3) and the range constrained CCA (4) can be completely described and robustly implemented via the SVD, the maximizers P_a and P_b do not have orthonormal columns. Naturally one would think of simply postprocessing the obtained P_a and P_b by orthogonalizing their columns, respectively. Evidence suggests that the resulting P_a and P_b may not be effective for tasks that follow [5], also as our later numerical experiments will show.

In 2013, Shen *et al.* [24] proposed the following orthogonal CCA (OCCA) model, hereafter called *Shen-Sun-Yuan OCCA model*, to define a transformation orthogonal matrix pair (P_a, P_b) as a set of orthonormal vector pairs $(p_{a\ell}, p_{b\ell})$, one at a time. The basic idea is very much similar to the SVD which can be constructed sequentially one singular triplet at a time.

Shen-Sun-Yuan OCCA model:

(1) Define the first pair, denoted by (p_{a1}, p_{b1}), as the pair of maximizers of

$$\max_{\|p_a\|_2=\|p_b\|_2=1} \frac{p_a^{\mathrm{T}} C p_b}{\sqrt{(p_a^{\mathrm{T}} A p_a)(p_b^{\mathrm{T}} B p_b)}},$$

i.e. (3a) with $k = 1$.

(2) Suppose that (p_{ai}, p_{bi}) for $1 \leq i \leq \ell - 1$ have already been defined. The next pair $(p_{a\ell}, p_{b\ell})$, normalized to have $\|p_{a\ell}\|_2 = \|p_{b\ell}\|_2 = 1$, is defined as the pair of maximizers of

$$\max_{\|p_a\|_2=\|p_b\|_2=1} \frac{p_a^{\mathrm{T}} C p_b}{\sqrt{(p_a^{\mathrm{T}} A p_a)(p_b^{\mathrm{T}} B p_b)}}$$
$$\text{subject to} \quad p_a^{\mathrm{T}} P_{a\,(\ell-1)} = 0,\ p_b^{\mathrm{T}} P_{b\,(\ell-1)} = 0,$$

where $P_{a\,(\ell-1)} = [p_{a1}, p_{a2}, \ldots, p_{a\,(\ell-1)}]$ and $P_{b\,(\ell-1)} = [p_{b1}, p_{b2}, \ldots, p_{b\,(\ell-1)}]$.

(3) The process is repeated as necessary.

Remark 2.1: In [24], both maximization problems in Steps 1 and 2 above were stated with the constraint $p_a^{\mathrm{T}}Ap_a = p_b^{\mathrm{T}}Bp_b = 1$, instead of $\|p_a\|_2 = \|p_b\|_2 = 1$. This discrepancy is mathematically inconsequential because of the objective function

$$\rho(p_a, p_b) := \frac{p_a^{\mathrm{T}}Cp_b}{\sqrt{(p_a^{\mathrm{T}}Ap_a)(p_b^{\mathrm{T}}Bp_b)}}$$

is invariant with respect to the transformations $p_a \leftarrow \alpha p_a$ and $p_b \leftarrow \beta p_b$ for any positive scalars $\alpha,\ \beta \in \mathbb{R}$. Consider three possible constraints on p_a and p_b, respectively:

$$p_a^{\mathrm{T}}Ap_a = 1, \quad \|p_a\|_2 = 1, \quad p_a \neq 0, \tag{10a}$$

$$p_b^{\mathrm{T}}Bp_b = 1, \quad \|p_b\|_2 = 1, \quad p_b \neq 0. \tag{10b}$$

Maximizing the quotient $\rho(p_a, p_b)$ subject to p_a satisfying any one of the constraints in (10a) and p_b satisfying any one of constraints in (10b) (besides $p_a^{\mathrm{T}}P_{a\,(\ell-1)} = 0$, $p_b^{\mathrm{T}}P_{b\,(\ell-1)} = 0$ in Step 2) yield the same optimal p_a and p_b in direction. Here we pick the constraint $\|p_a\|_2 = \|p_b\|_2 = 1$ to avoid explicit normalization after optimal p_a and p_b in direction are determined.

A numerical method is also proposed in [24] to realize this OCCA model. They used the SVD approach as outlined in Section 2.1 for computing the first pair (p_{a1}, p_{b1}). For the subsequent pairs $(p_{a\ell}, p_{b\ell})$, they established a theorem [24, Theorem 1] which says that $(p_{a\ell}, p_{b\ell})$ can be recovered from the eigenvector associated with the largest eigenvalue of a nonsymmetric eigenvalue problem. It is this step that is most problematic computationally because, as we attempted to repeat the numerical results in [24], we found that the *largest eigenvalue* of the nonsymmetric eigenvalue problem in [24, Theorem 1] may not be well-defined. Numerically, we (and the authors of [24] themselves too[1]) have often encountered that the largest many eigenvalues (in magnitude) are complex. That leads to an impasse with no sensible way to go forward.

3. Range constrained orthogonal CCA

Drawing inspiration from the range constrained CCA (4) as an improved variant of CCA (3), we propose yet a new OCCA model as a variant of the Shen-Sun-Yuan OCCA model, by adding range constraints.

Range constrained OCCA model:

(1) Define the first pair, denoted by (p_{a1}, p_{b1}), as the pair of maximizers of

$$\max_{\|p_a\|_2=\|p_b\|_2=1} \frac{p_a^{\mathrm{T}}Cp_b}{\sqrt{(p_a^{\mathrm{T}}Ap_a)(p_b^{\mathrm{T}}Bp_b)}}$$
$$\text{subject to} \quad p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(X_b),$$

i.e. (4) with $k = 1$.

(2) Suppose that (p_{ai}, p_{bi}) for $1 \le i \le \ell - 1$ have already been defined. The next pair $(p_{a\ell}, p_{b\ell})$ is defined as the pair of maximizers of

$$\max_{\|p_a\|_2=\|p_b\|_2=1} \frac{p_a^{\mathrm{T}}Cp_b}{\sqrt{(p_a^{\mathrm{T}}Ap_a)(p_b^{\mathrm{T}}Bp_b)}} \tag{11a}$$

$$\text{subject to} \quad p_a^{\mathrm{T}} P_{a(\ell-1)} = 0,\ p_b^{\mathrm{T}} P_{b(\ell-1)} = 0, \tag{11b}$$

$$p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(X_b), \tag{11c}$$

where $P_{a(\ell-1)} = [p_{a1}, p_{a2}, \ldots, p_{a(\ell-1)}]$ and $P_{b(\ell-1)} = [p_{b1}, p_{b2}, \ldots, p_{b(\ell-1)}]$.

(3) The process is repeated as many times as needed.

To numerically realize the model, for Step 1, we can use the SVD approach that we outlined above from (5a) to (9) to compute (p_{a1}, p_{b1}). To solve (11a), similar to the deflation idea for the sparse PCA [20], we first establish Lemma 3.1 and Theorem 3.1.

Lemma 3.1: *Let $W_a \in \mathbb{R}^{n\times t}$, and suppose that $W_a^{\mathrm{T}} W_a = I_t$ and $\mathcal{R}(W_a) \subseteq \mathcal{R}(X_a)$. Then $p_a^{\mathrm{T}} W_a = 0$ and $p_a \in \mathcal{R}(X_a)$ if and only if $p_a \in \mathcal{R}(\widetilde{X}_a)$, where*

$$\widetilde{X}_a = (I - W_a W_a^{\mathrm{T}}) X_a. \tag{12}$$

Proof: Notice $\mathcal{R}(W_a) \subseteq \mathcal{R}(X_a)$ to get

$$\mathcal{R}(\widetilde{X}_a) = \mathcal{R}(X_a - W_a W_a^{\mathrm{T}} X_a) \subseteq \mathcal{R}(X_a)$$

and notice $W_a^{\mathrm{T}} W_a = I_t$ to get

$$W_a^{\mathrm{T}} \widetilde{X}_a = (W_a^{\mathrm{T}} - W_a^{\mathrm{T}} W_a W_a^{\mathrm{T}}) X_a = (W_a^{\mathrm{T}} - W_a^{\mathrm{T}}) X_a = 0.$$

Therefore, if $p_a \in \mathcal{R}(\widetilde{X}_a)$, then $p_a^{\mathrm{T}} W_a = 0$ and $p_a \in \mathcal{R}(X_a)$. On the other hand, suppose $p_a^{\mathrm{T}} W_a = 0$ and $p_a \in \mathcal{R}(X_a)$. By $p_a^{\mathrm{T}} W_a = 0$, we have $W_a^{\mathrm{T}} p_a = 0 \Rightarrow W_a W_a^{\mathrm{T}} p_a = 0$. Therefore

$$p_a = p_a - W_a W_a^{\mathrm{T}} p_a = (I - W_a W_a^{\mathrm{T}}) p_a.$$

Since also $p_a \in \mathcal{R}(X_a)$, there exists a vector z_a such that $p_a = X_a z_a$. Hence

$$p_a = (I - W_a W_a^{\mathrm{T}}) X_a z_a = \widetilde{X}_a z_a \in \mathcal{R}(\widetilde{X}_a),$$

as expected. ■

Theorem 3.1: *Let $W_a \in \mathbb{R}^{n\times t}$, $W_b \in \mathbb{R}^{m\times t}$. Suppose that $W_a^{\mathrm{T}} W_a = W_b^{\mathrm{T}} W_b = I_t$, $\mathcal{R}(W_a) \subseteq \mathcal{R}(X_a)$ and $\mathcal{R}(W_b) \subseteq \mathcal{R}(X_b)$. Then the following maximization problem*

$$\max_{\|p_a\|_2 = \|p_b\|_2 = 1} \quad \frac{p_a^{\mathrm{T}} C p_b}{\sqrt{(p_a^{\mathrm{T}} A p_a)(p_b^{\mathrm{T}} B p_b)}} \tag{13a}$$

$$\text{subject to} \quad p_a^{\mathrm{T}} W_a = 0,\ p_b^{\mathrm{T}} W_b = 0, \tag{13b}$$

$$p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(X_b), \tag{13c}$$

is equivalent to

$$\max_{\|p_a\|_2 = \|p_b\|_2 = 1} \quad \frac{p_a^{\mathrm{T}} \widetilde{C} p_b}{\sqrt{(p_a^{\mathrm{T}} \widetilde{A} p_a)(p_b^{\mathrm{T}} \widetilde{B} p_b)}} \tag{14a}$$

$$\text{subject to} \quad p_a \in \mathcal{R}(\widetilde{X}_a), \quad p_b \in \mathcal{R}(\widetilde{X}_b), \tag{14b}$$

in the sense that both have the same pair of maximizers, where $\widetilde{X}_a = (I - W_a W_a^{\mathrm{T}})X_a$, $\widetilde{X}_b = (I - W_b W_b^{\mathrm{T}})X_b$, $\widetilde{A} = \widetilde{X}_a\widetilde{X}_a^{\mathrm{T}}$, $\widetilde{B} = \widetilde{X}_b\widetilde{X}_b^{\mathrm{T}}$ *and* $\widetilde{C} = \widetilde{X}_a\widetilde{X}_b^{\mathrm{T}}$.

Proof: By Lemma 3.1, we see that the constraints of (13) and these of (14) are the same. Next, For (p_a, p_b) satisfying the constraints, we have

$$\widetilde{X}_a^{\mathrm{T}} p_a = X_a^{\mathrm{T}}(I - W_a W_a^{\mathrm{T}})^{\mathrm{T}} p_a = X_a^{\mathrm{T}}(p_a - W_a W_a^{\mathrm{T}} p_a) = X_a^{\mathrm{T}} p_a, \quad \widetilde{X}_b^{\mathrm{T}} p_b = X_b^{\mathrm{T}} p_b.$$

Therefore, $p_a^{\mathrm{T}} A p_a = p_a^{\mathrm{T}}\widetilde{A}p_a$, $p_b^{\mathrm{T}} B p_b = p_b^{\mathrm{T}}\widetilde{B}p_b$, and $p_a^{\mathrm{T}} C p_b = p_a^{\mathrm{T}}\widetilde{C}p_b$, implying the two objective functions have the same value. ■

The importance of Theorem 3.1 is that it relates the maximization problem (13) to (14). The pair of maximizers of (14) can be again solved robustly by the SVD approach outlined from (5a) to (9) in Section 2.1. Since the range constrained OCCA problem (11) is in the form of (13) with $W_a = P_{a\,(\ell-1)}$ and $W_b = P_{b\,(\ell-1)}$, the problem (11) can be solved robustly by the SVD approach. In actual implementation, we may overwrite X_a and X_b for $\widetilde{X}_a$ and $\widetilde{X}_b$ and update them one vector at time as soon as a new pair is computed. We summarize the method for solving the the range constrained OCCA model in Algorithm 1.

Algorithm 1 rc-OCCA: range constrained OCCA

Require: $X_a \in \mathbb{R}^{n\times q}$ and $X_b \in \mathbb{R}^{m\times q}$ (both centred and, optionally, columns normalized), integer $1 \le k \le \min\{m, n, q\}$;

Ensure: $P_a = [p_{a1}, \ldots, p_{ak}] \in \mathbb{R}^{n\times k}$ and $P_b = [p_{b1}, \ldots, p_{bk}] \in \mathbb{R}^{m\times k}$ that solve the range constrained OCCA model and satisfy $P_a^{\mathrm{T}} P_a = P_b^{\mathrm{T}} P_b = I_k$.

1: compute the SVDs in (5);
2: compute the SVD (8), and let $p_{a1} = U_a \Sigma_a^{-1} U_{(:,1)}$, $p_{b1} = U_b \Sigma_b^{-1} V_{(:,1)}$;
3: **for** $i = 2$ to k **do**
4: $\quad X_a = X_a - p_{a\,(i-1)}(p_{a\,(i-1)}^{\mathrm{T}} X_a)$, $X_b = X_b - p_{b\,(i-1)}(p_{b\,(i-1)}^{\mathrm{T}} X_b)$;
5: $\quad$ compute the SVDs in (5);
6: $\quad$ compute the SVD (8), and let $p_{ai} = U_a \Sigma_a^{-1} U_{(:,1)}$, $p_{bi} = U_b \Sigma_b^{-1} V_{(:,1)}$;
7: **end for**
8: **return** $P_a = [p_{a1}, \ldots, p_{ak}]$ and $P_b = [p_{b1}, \ldots, p_{bk}]$.

4. Range constrained partial OCCA

In the OCCA models of the previous section, both transformation matrices P_a and P_b have orthonormal columns. In some applications [25,26], datasets X_a and X_b stand for source input and target, respectively. For example, in multi-label classification [26] (see Section 5.3), X_a is the input data, while X_b is the target class labels where each entry is generally represented by a binary variable: '1' for the existence of one label and '0' for non-existence of the label. Since one sample is allowed to have multiple labels, the orthogonal basis in P_b for X_b might not be able to fully characterize the correlations among labels. Hence, there are needs and justifications to make only one of them, say P_a, have orthonormal columns, while $P_b^{\mathrm{T}} B P_b = I_k$ for P_b because of the needs in keeping the correlations of the target labels. Collectively, we will call such variants *partial OCCA* (pOCCA).

Range constrained pOCCA model:

(1) Define the first pair, denoted by (p_{a1}, p_{b1}), as the pair of maximizers of

$$\max_{\|p_a\|_2=\|p_b\|_B=1} \quad \frac{p_a^{\mathrm{T}} C p_b}{\sqrt{(p_a^{\mathrm{T}} A p_a)(p_b^{\mathrm{T}} B p_b)}}$$
$$\text{subject to} \quad p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(X_b)$$

i.e. (29) with $k = 1$.

(2) Suppose that (p_{ai}, p_{bi}) for $1 \le i \le \ell - 1$ have already been computed. The next pair $(p_{a\ell}, p_{b\ell})$ is defined as the pair of maximizers of

$$\max_{\|p_a\|_2=\|p_b\|_B=1} \quad \frac{p_a^{\mathrm{T}} \widehat{C} p_b}{\sqrt{(p_a^{\mathrm{T}} A p_a)(p_b^{\mathrm{T}} \widetilde{B} p_b)}} \tag{15a}$$
$$\text{subject to} \quad p_a^{\mathrm{T}} P_{a(\ell-1)} = 0, \tag{15b}$$
$$p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(\widetilde{X}_b), \tag{15c}$$

where $P_{a(\ell-1)} = [p_{a1}, p_{a2}, \ldots, p_{a(\ell-1)}]$, $P_{b(\ell-1)} = [p_{b1}, p_{b2}, \ldots, p_{b(\ell-1)}]$, and

$$\widetilde{X}_b = [I - P_{b(\ell-1)} P_{b(\ell-1)}^{\mathrm{T}} B] X_b, \quad \widetilde{B} = \widetilde{X}_b \widetilde{X}_b^{\mathrm{T}}, \quad \widehat{C} = X_a \widetilde{X}_b.$$

(3) The process is repeated as necessary.

The range constrained pOCCA model will define the columns of both P_a and P_b sequentially. It is clear that at the end $P_a^{\mathrm{T}} P_a = I_k$, but it is not so clear if $P_b^{\mathrm{T}} B P_b = I_k$ also. In fact, it is, as a corollary of the following lemma which says $p_b \in \mathcal{R}(\widetilde{X}_b)$ in (15) implies $p_b^{\mathrm{T}} B P_{b(\ell-1)} = 0$ and $p_b \in \mathcal{R}(X_b)$.

Lemma 4.1: *Let $W_b \in \mathbb{R}^{n\times t}$, and suppose that $W_b^{\mathrm{T}} B W_b = I_t$ and $\mathcal{R}(W_b) \subseteq \mathcal{R}(X_b)$. Then $p_b^{\mathrm{T}} B W_b = 0$ and $p_b \in \mathcal{R}(X_b)$ if and only if $p_b \in \mathcal{R}(\widetilde{X}_b)$, where*

$$\widetilde{X}_b = (I - W_b W_b^{\mathrm{T}} B) X_b. \tag{16}$$

Proof: Since $\mathcal{R}(W_b) \subseteq \mathcal{R}(X_b)$, we find $\mathcal{R}(\widetilde{X}_b) \subseteq \mathcal{R}(X_b)$. It can also be verified that $(BW_b)^{\mathrm{T}} \widetilde{X}_b = 0$ upon using $W_b^{\mathrm{T}} B W_b = I_t$. Thus if $p_b \in \mathcal{R}(\widetilde{X}_b)$, then $p_b^{\mathrm{T}} B W_b = 0$ and $p_b \in \mathcal{R}(X_b)$. On other hand, if $p_b^{\mathrm{T}} B W_b = 0$ and $p_b \in \mathcal{R}(X_b)$, then

$$p_b = p_b - W_b W_b^{\mathrm{T}} B p_b = (I - W_b W_b^{\mathrm{T}} B) p_b = (I - W_b W_b^{\mathrm{T}} B) X_b z_b$$

for some vector z_b. Hence $p_b \in \mathcal{R}(\widetilde{X}_b)$. ■

To numerically realize the range constrained pOCCA model, for Step 1, we can again use the SVD approach that we outlined above from (5) to (9) to compute (p_{a1}, p_{b1}). To solve (15), we establish the following theorem similar to Theorem 3.1.

Theorem 4.1: *Let $W_a \in \mathbb{R}^{n\times t}$, $W_b \in \mathbb{R}^{m\times t}$. Suppose that $W_a^{\mathrm{T}} W_a = W_b^{\mathrm{T}} B W_b = I_t$, $\mathcal{R}(W_a) \subseteq \mathcal{R}(X_a)$ and $\mathcal{R}(W_b) \subseteq \mathcal{R}(X_b)$. Then the following maximization problem*

$$\max_{\|p_a\|_2=\|p_b\|_B=1} \quad \frac{p_a^{\mathrm{T}} \widehat{C} p_b}{\sqrt{(p_a^{\mathrm{T}} A p_a)(p_b^{\mathrm{T}} \widetilde{B} p_b)}} \tag{17a}$$
$$\text{subject to} \quad p_a^{\mathrm{T}} W_a = 0, \tag{17b}$$

$$p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(\widetilde{X}_b), \tag{17c}$$

is equivalent to

$$\max_{\|p_a\|_2=\|p_b\|_B=1} \quad \frac{p_a^{\mathrm{T}}\widetilde{C}p_b}{\sqrt{(p_a^{\mathrm{T}}\widetilde{A}p_a)(p_b^{\mathrm{T}}\widetilde{B}p_b)}} \tag{18a}$$

$$\text{subject to} \quad p_a \in \mathcal{R}(\widetilde{X}_a), \quad p_b \in \mathcal{R}(\widetilde{X}_b), \tag{18b}$$

in the sense that both have the same pair of maximizers, where $\widetilde{X}_a$ and $\widetilde{X}_b$ are as defined in (12) and (16), respectively, and $\widetilde{A} = \widetilde{X}_a\widetilde{X}_a^{\mathrm{T}}, \widetilde{B} = \widetilde{X}_b\widetilde{X}_b^{\mathrm{T}}, \widehat{C} = X_a\widetilde{X}_b^{\mathrm{T}}$, and $\widetilde{C} = \widetilde{X}_a\widetilde{X}_b^{\mathrm{T}}$.

Proof: By Lemmas 3.1 and 4.1, we see that the constraints of the problem (17) and those of (18) are the same. Next for (p_a, p_b) that satisfies the constraints, we have $\widetilde{X}_a^{\mathrm{T}}p_a = X_a^{\mathrm{T}}p_a$. Therefore $p_a^{\mathrm{T}}\widetilde{A}p_a = p_a^{\mathrm{T}}Ap_a$ and $p_a^{\mathrm{T}}\widehat{C}p_b = p_a^{\mathrm{T}}\widetilde{C}p_b$. ■

The optimization problem (18) can be solved in the same way as we did for Step 1. In actual implementation, we may overwrite X_a and X_b for $\widetilde{X}_a$ and $\widetilde{X}_b$ and update them one vector at time as soon as a new pair is computed. We summarize this method in Algorithm 2.

Algorithm 2 rc-pOCCA: range constrained pOCCA

Require: $X_a \in \mathbb{R}^{n\times q}$ and $X_b \in \mathbb{R}^{m\times q}$ (both centred and, optionally, columns normalized), integer $1 \le k \le \min\{m, n, q\}$;

Ensure: $P_a = [p_{a1}, \ldots, p_{ak}] \in \mathbb{R}^{n\times k}$ and $P_b = [p_{b1}, \ldots, p_{bk}] \in \mathbb{R}^{m\times k}$ that solve the range constrained pOCCA model and satisfy $P_a^{\mathrm{T}}P_a = P_b^{\mathrm{T}}BP_b = I_k$.

1: $B = X_bX_b^{\mathrm{T}}$;
2: compute the SVDs in (5);
3: compute the SVD (8), and let $p_{a1} = U_a\Sigma_a^{-1}U_{(:,1)}$, $\hat{p}_{b1} = U_b\Sigma_b^{-1}V_{(:,1)}$, $p_{b1} = \hat{p}_{b1}/\|\hat{p}_{b1}\|_B$;
4: **for** $i = 2$ to k **do**
5: $\quad X_a = X_a - p_{a\,(i-1)}(p_{a\,(i-1)}^{\mathrm{T}}X_a)$, $X_b = X_b - p_{b\,(i-1)}((p_{b\,(i-1)}^{\mathrm{T}}B)X_b)$;
6: $\quad$ compute the SVDs in (5);
7: $\quad$ compute the SVD (8), and let $p_{ai} = U_a\Sigma_a^{-1}U_{(:,1)}$, $\hat{p}_{bi} = U_b\Sigma_b^{-1}V_{(:,1)}$, $p_{bi} = \hat{p}_{bi}/\|\hat{p}_{bi}\|_B$;
8: **end for**
9: **return** $P_a = [p_{a1}, \ldots, p_{ak}]$ and $P_b = [p_{b1}, \ldots, p_{bk}]$.

Remark 4.1: We would like to draw the attention of the reader to the objective function in (15) of the range constrained pOCCA model. It is perhaps not in the form as one might expect. The objective function (15) is designed in such a way so that the resulting maximization problem is readily solvable by the SVD approach while at the end $P_b^{\mathrm{T}}BP_b = I_k$ is guaranteed. In view of the range constrained OCCA model in Section 2.2, it is perhaps more natural in Step 2 of the range constrained pOCCA model to use, instead of (15),

$$\max_{\|p_a\|_2=\|p_b\|_B=1} \quad \frac{p_a^{\mathrm{T}}Cp_b}{\sqrt{(p_a^{\mathrm{T}}Ap_a)(p_b^{\mathrm{T}}Bp_b)}} \tag{19a}$$

$$\text{subject to} \quad p_a^{\mathrm{T}} P_{a(\ell-1)} = 0, \quad p_b^{\mathrm{T}} B P_{b(\ell-1)} = 0, \tag{19b}$$

$$p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(X_b). \tag{19c}$$

Indeed, our first try was precisely this, and then we found that there is no good way to solve it by the SVD. We now explain. To simplify notation, we consider

$$\max_{\|p_a\|_2=\|p_b\|_B=1} \quad \frac{p_a^{\mathrm{T}} C p_b}{\sqrt{(p_a^{\mathrm{T}} A p_a)(p_b^{\mathrm{T}} B p_b)}} \tag{20a}$$

$$\text{subject to} \quad p_a^{\mathrm{T}} W_a = 0, \quad p_b^{\mathrm{T}} B W_b = 0, \tag{20b}$$

$$p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(X_b), \tag{20c}$$

where $W_a \in \mathbb{R}^{n\times t}$, $W_b \in \mathbb{R}^{m\times t}$, satisfying $W_a^{\mathrm{T}} W_a = W_b^{\mathrm{T}} B W_b = I_t$, $\mathcal{R}(W_a) \subseteq \mathcal{R}(X_a)$ and $\mathcal{R}(W_b) \subseteq \mathcal{R}(X_b)$. By Lemmas 3.1 and 4.1, we see (20c) is equivalent to $p_a \in \mathcal{R}(\widetilde{X}_a)$ and $p_b \in \mathcal{R}(\widetilde{X}_b)$, where $\widetilde{X}_a$ and $\widetilde{X}_b$ are as defined in (12) and (16), respectively. As in the proof of Theorem 3.1, for $p_a \in \mathcal{R}(\widetilde{X}_a)$ we have $\widetilde{X}_a^{\mathrm{T}} p_a = X_a^{\mathrm{T}} p_a$, but for $p_b \in \mathcal{R}(\widetilde{X}_b)$, we do not have $\widetilde{X}_b^{\mathrm{T}} p_b = X_b^{\mathrm{T}} p_b$. In fact, since $p_b \in \mathcal{R}(\widetilde{X}_b)$ implies $(I - W_b W_b^{\mathrm{T}} B) p_b = p_b$, we have

$$X_b^{\mathrm{T}} p_b = X_b^{\mathrm{T}} (I - W_b W_b^{\mathrm{T}} B) p_b =: \widehat{X}_b^{\mathrm{T}} p_b,$$

where $\widehat{X}_b = (I - W_b W_b^{\mathrm{T}} B)^{\mathrm{T}} X_b = (I - B W_b W_b^{\mathrm{T}}) X_b$. Consequently, the maximization problem (20) is equivalent to

$$\max_{\|p_a\|_2=\|p_b\|_B=1} \quad \frac{p_a^{\mathrm{T}} \widetilde{X}_a \widehat{X}_b^{\mathrm{T}} p_b}{\sqrt{(p_a^{\mathrm{T}} \widetilde{A} p_a)(p_b^{\mathrm{T}} \widehat{B} p_b)}} \tag{21a}$$

$$\text{subject to} \quad p_a \in \mathcal{R}(X_a), \quad p_b \in \mathcal{R}(\widetilde{X}_b), \tag{21b}$$

where $\widehat{B} = \widehat{X}_b \widehat{X}_b^{\mathrm{T}}$. In general, $\widehat{X}_b = (I - B W_b W_b^{\mathrm{T}}) X_b \neq \widetilde{X}_b = (I - W_b W_b^{\mathrm{T}} B) X_b$, and thus $\widehat{X}_b^{\mathrm{T}} p_b \neq \widetilde{X}_b^{\mathrm{T}} p_b$. But more seriously, $\mathcal{R}(\widehat{X}_b) \neq \mathcal{R}(\widetilde{X}_b)$, making the SVD approach that we outlined above from (5) to (9) not readily suitable to solve (21).

5. Numerical experiments

We have conducted extensive numerical experiments to evaluate the proposed range constrained OCCA and pOCCA models as feature extraction approaches on three popular applications of CCA, namely unsupervised feature fusion, multi-target regression and multi-label classification.

5.1. Unsupervised feature fusion

Feature fusion is an important part of information fusion. Multiple features can be extracted from the same pattern, and they usually reflect different characteristics of the pattern. The aim of the feature fusion is to combine different sets of features for better classification. CCA was used for feature fusion by effectively leveraging the inherent correlations between two feature sets [27]. Specifically, the CCA-based fusion method first

extracts canonical correlation features from two groups of feature vectors of the same pattern; and then a fusion method is used to combine two sets of canonical correlation features; and finally, the fused features are used for pattern recognition.

We evaluate our proposed range constrained OCCA models on the real-world datasets with the inputs of two feature matrices X_a and X_b extracted from the same patterns by comparing with the traditional CCA. It is worth noting that the supervised information such as class labels are not used for inferring the fused features. To evaluate the performance, we split the dataset into training set and testing set, denoted by (X_a, X_b) and (X'_a, X'_b), respectively. The training stage and testing stage of CCA-based unsupervised feature fusion are shown as follows.

Training Stage

(1) Input datasets $X_a \in \mathbb{R}^{n\times q}$ and $X_b \in \mathbb{R}^{m\times q}$
(2) Centralize the datasets $X_a \leftarrow X_a - \mu_a \mathbf{1}_q^{\mathrm{T}}$ and $X_b \leftarrow X_b - \mu_b \mathbf{1}_q^{\mathrm{T}}$, where centres $\mu_a = (1/q)X_a\mathbf{1}_q$ and $\mu_b = (1/q)X_b\mathbf{1}_q$.
(3) Apply the range constrained CCA or OCCA model to the dataset pair (X_a, X_b) and obtain transformation matrix pair (P_a, P_b), for a selected positive integer k $(1 \le k \le \min\{m, n, p\})$.
(4) Use two fusion strategies, namely serial feature fusion (denoted as PR1) and parallel feature fusion (denoted as PR2), to obtain a fused feature matrix Z:

$$\text{(PR1)} \quad Z = \begin{bmatrix} P_a^{\mathrm{T}} & 0 \\ 0 & P_b^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \end{bmatrix} = \begin{bmatrix} P_a^{\mathrm{T}} X_a \\ P_b^{\mathrm{T}} X_b \end{bmatrix}, \tag{22}$$

$$\text{(PR2)} \quad Z = \begin{bmatrix} P_a \\ P_b \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} X_a \\ X_b \end{bmatrix} = P_a^{\mathrm{T}} X_a + P_b^{\mathrm{T}} X_b. \tag{23}$$

Testing and Evaluation Stage

(1) Input datasets $X'_a \in \mathbb{R}^{n\times q'}$ and $X'_b \in \mathbb{R}^{m\times q'}$, transformation matrices P_a and P_b, centralized vectors μ_a and μ_b, and class labels $y \in \{1, 2, \ldots, c\}^q$ of the training dataset and $y' \in \{1, 2, \ldots, c\}^{q'}$ of the testing dataset, where c is the number of classes.
(2) Centralize testing datasets using centres from the training set: $X'_a \leftarrow X'_a - \mu_a \mathbf{1}_{q'}^{\mathrm{T}}$ and $X'_b \leftarrow X'_b - \mu_b \mathbf{1}_{q'}^{\mathrm{T}}$.
(3) Apply the fusion strategies (22) or (23) to obtain a fused feature matrix Z'.
(4) Compute the prediction using the nearest neighbour classifier, for $i = 1, \ldots, q'$,

$$i^* = \arg\min_{j=1,\ldots,q} \|z'_i - z_j\|_2, \quad y_i^* = y_{i^*}, \tag{24}$$

where z_j is the jth column of Z and z'_i is the ith column of Z'.
(5) Given true classes y', the classification accuracy is computed as

$$\text{accuracy}(y', y^*) = \frac{1}{q'} \sum_{i=1}^{q'} \delta(y'_i, y_i^*), \tag{25}$$

where $\delta(y'_i, y_i^*) = 1$ if $y'_i = y_i^*$ and 0 otherwise.

We evaluate the classification performance of each method via the nearest neighbour classifier over the testing data in terms of the accuracy (25). The bigger the accuracy is, the better the method performs.

A publicly available multiple feature dataset[2] is used for the classification evaluation of the compared methods. It consists of features of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. Hence, there are $c = 10$ classes, each of which has

200 patterns, so there are $q + q' = 2000$ in total. Each pattern was digitized as a binary image and is represented by 6 different feature sets:

(1) fou: 76 Fourier coefficients of the character shapes;
(2) fac: 216 profile correlations;
(3) kar: 64 Karhunen-Love coefficients;
(4) pix: 240 pixel averages in 2×3 windows;
(5) zer: 47 Zernike moments;
(6) mor: 6 morphological features.

Each method takes two sets of features as the inputs, and so there are 15 combinations. For each combination, we randomly choose $q = 300$ samples for training, and the rest $q' = 1700$ samples are used for testing. As the minimum number of features in the six views is 6, we set parameter k of OCCA the same as the maximal dimension obtained by CCA. Each random experiment is repeated 10 times and the average results in terms of classification accuracy are reported.

In Table 1, we show the average classification accuracies of the traditional CCA model (3) via MATLAB's `canoncorr` and the range constrained OCCA model (Algorithm 1) by using both fusion methods `PR1` and `PR2`. From Table 1, we have the following observations:

(1) The range constrained OCCA model (Algorithm 1) has achieved significantly improved recognition rates for all 15 testing combinations in terms of both the serial feature strategy (`PR1`) and the parallel feature strategy (`PR2`), over all 15 combinations. A major reason is that the basis constructed by the traditional CCA is only conjugate orthogonal, which is heavily influenced by both the number and the dimensionality of samples, while a directly constructed orthogonal basis seems less sensitive to them.

Table 1. Average classification accuracies of unsupervised feature fusion by the range constrained traditional CCA (4) and OCCA (11).

		CCA		OCCA	
No.	Feature combination	PR1	PR2	PR1	PR2
1	Fou-Fac	0.5739	0.5568	**0.9581**	**0.9413**
2	Fou-Kar	0.7838	0.7223	**0.9596**	**0.9390**
3	Fou-Pix	0.4482	0.4461	**0.9599**	**0.9434**
4	Fou-Zer	0.7183	0.6931	**0.8482**	**0.8189**
5	Fou-Mor	0.7052	0.6918	**0.8254**	**0.7326**
6	Fac-Kar	0.8131	0.8039	**0.9488**	**0.9295**
7	Fac-Pix	0.5675	0.5331	**0.9481**	**0.9399**
8	Fac-Zer	0.7128	0.7113	**0.9310**	**0.9215**
9	Fac-Mor	0.6531	0.6499	**0.9178**	**0.7984**
10	Kar-Pix	0.7827	0.7800	**0.9298**	**0.9280**
11	Kar-Zer	0.8362	0.8042	**0.9472**	**0.8542**
12	Kar-Mor	0.7519	0.7280	**0.9441**	**0.8648**
13	Pix-Zer	0.5884	0.5914	**0.9522**	**0.8760**
14	Pix-Mor	0.5367	0.5231	**0.9449**	**0.8895**
15	Zer-Mor	0.7109	0.6952	**0.7788**	**0.7182**

(2) The rc-OCCA model using the serial feature strategy (`PR1`) generally shows better results than using the parallel feature strategy (`PR2`).

Based on the above observations, we may reasonably draw conclusions that the OCCA outperforms the traditional CCA, and is more robust, accurate, and effective.

We note that since Shen-Sun-Yuan OCCA model and algorithm [24] encounter numerical difficulty due to the involvement of complex eigenvalues and eigenvectors of nonsymmetric matrix eigenvalue problems, we cannot replicate their numerical experiments and, consequently, we do not include their numerical results.

5.2. Multi-Target regression

Multi-target regression (MTR) is the task of predicting multiple continuous variables using a common set of input variables [25]. Such problems arise in various fields such as ecological modeling, economics and energy. An informal definition of the MTR task can be explained as follows. Let $X_a \in \mathbb{R}^{n\times q}$ be the input data and $X_b \in \mathbb{R}^{m\times q}$ be the corresponding output, where n is the number of features, m is the number of output targets, and q is the number of data points. For each input data point x_a, its corresponding output target is x_b. We assume that a sample (x_a, x_b) is identically and independently sampled from a joint unknown distribution. Given a set of q sample pairs consisting of corresponding columns of X_a and X_b, the goal of MTR is to learn a function h that is able to predict, for given input x_a, an output $\widehat{x}_b = h(x_a)$ that best approximates the true output x_b with good generalization to unseen data or testing data. Following the conventional method [25], MTR is transformed into a series of single-target regression problems. Componentwise, we write $h = [h_1, \ldots, h_m]^{\mathrm{T}}$. In what follows, we summarize the training stage and testing stage for CCA-based MTR.

Training stage

(1) Input datasets $X_a \in \mathbb{R}^{n\times q}$ and $X_b \in \mathbb{R}^{m\times q}$.
(2) Centralize the datasets $X_a \leftarrow X_a - \mu_a \mathbf{1}_q^{\mathrm{T}}$ and $X_b \leftarrow X_b - \mu_b \mathbf{1}_q^{\mathrm{T}}$, where the centres $\mu_a = (1/q)X_a\mathbf{1}_q$ and $\mu_b = (1/q)X_b\mathbf{1}_q$.
(3) Apply the range constrained CCA, OCCA or pOCCA models to the dataset pair (X_a, X_b) and obtain transformation matrix pair (P_a, P_b) for a chosen positive integer k $(1 \le k \le \min\{m, n, q\})$. Note that for the pOCCA model, orthogonal constraint is only imposed on P_a.
(4) Define the transformed data $Z = P_a^{\mathrm{T}} X_a$, which is considered as the most correlated to target.
(5) Use the L_2-regularized L_2-loss support vector regression model [9] to learn the linear function $h(z) = w^{\mathrm{T}}z + b$ from any input z to its output target $h(z)$. Specifically, for the transformed data $Z = [z_1, \ldots, z_q]$ and its true target matrix X_b with the jth target of the ith data point denoted by $(X_b)_{j,i}$, solve the following optimization problem with respect to $W = [w_1, \ldots, w_m] \in \mathbb{R}^{n\times m}$ and $b = [b_1, \ldots, b_m] \in \mathbb{R}^m$:

$$\min_{W,b} \sum_{j=1}^{m} \left[\frac{1}{2} w_j^{\mathrm{T}} w_j + \alpha \sum_{i=1}^{q} \Big(\max\{0, |(X_b)_{j,i} - w_j^{\mathrm{T}} z_i - b_j| - \epsilon\} \Big)^2 \right], \tag{26}$$

where α is the regularization parameter and ϵ is another parameter used to tolerate the deviation of predicted target to the true target within $[-\epsilon, \epsilon]$. The liblinear toolbox[3] is used to solve (26).

Testing and evaluation stage

(1) Input test datasets $X'_a \in \mathbb{R}^{n\times q'}$ and $X'_b \in \mathbb{R}^{m\times q'}$, transformation matrix P_a and centres μ_a and μ_b.

(2) Centralize testing data $X'_a \leftarrow X'_a - \mu_a \mathbf{1}^{\mathrm{T}}_{q'}$ and $X'_b \leftarrow X'_b - \mu_b \mathbf{1}^{\mathrm{T}}_{q'}$.

(3) Apply the learned function h to predict the target value of testing dataset X'_a, i.e. for $i = 1, \ldots, q'$ and $j = 1, \ldots, m$,

$$(X_*)_{j,i} = w_j^{\mathrm{T}} z'_i + b_j,$$

where $Z' = P_a^{\mathrm{T}} X'_a \equiv [z'_1, z'_2, \ldots, z'_{q'}]$.

(4) Compute the mean square error (MSE)

$$\mathrm{MSE}(X_*, X_b) = \frac{1}{q'} \sum_{i=1}^{q'} \sum_{j=1}^{m} \left[(X_*)_{j,i} - (X'_b)_{j,i}\right]^2.$$

We use MSE to measure the learning performance of MTR. Smaller error indicates better performance. Similarly to unsupervised feature fusion, we set parameter k to be the maximal number that can be obtained by CCA. We set $\epsilon = 0.1$ as the default value and tune the regularization parameter in the support vector regression in the grid $\{0.01, 0.1, 1, 10, 100\}$ for the best performance of all compared methods.

The datasets used for MTR are shown in Table 2. These datasets were collected from a variety of application domains and are publicly available[4]. Detailed information about these datasets are available at the corresponding references cited in Table 2. For each dataset, 70% of the data is randomly chosen as the training set and the rest 30% is used as the testing set. Each random experiment is repeated 10 times and the average results in terms of MSE for the testing data are reported to evaluate the learning generalization performance.

The last three columns in Table 2 are MSEs for 3 models on 11 datasets, where the best results are highlighted in bold. We observe that OCCA achieves smallest MSEs in 6 out of 11 datasets, while the other two methods manage to perform the best on 5 datasets albeit marginally. For the 6 examples that OCCA achieves best MSEs, OCCA gains significantly over the traditional CCA, and pOCCA also achieves very good performance on these examples over the traditional CCA. From these observations, it is clear that the orthogonal constraints that are imposed in OCCA and pOCCA are good things to do and indeed often result in superior performances.

Table 2. Datasets for testing MTR with MSEs by range constrained traditional CCA (4), OCCA (11) and pOCCA (15).

No.	Data	Samples	Features	Targets	CCA	OCCA	pOCCA
1	oes97 [25]	334	263	16	144.0861	**3**.6446	5.2188
2	oes10 [25]	403	298	16	155.6010	**3**.1584	5.2769
3	scm1d [25]	9803	280	16	2.5760	3.6233	**2**.5352
4	scm20d [25]	8966	61	16	**6**.6199	7.8394	7.2190
5	wq [7]	1060	16	14	12.9751	**12**.8588	12.8691
6	atp1d [25]	337	411	6	939.5027	**1**.3920	5.0789
7	atp7d [25]	296	411	6	311.7144	**2**.4815	3.9091
8	andro [15]	49	30	6	123.7381	**4**.0023	4.6754
9	slump [31]	103	7	3	**1**.5772	2.5342	1.5809
10	edm [19]	154	16	2	1.1502	1.2728	**1**.1292
11	enb [29]	768	8	2	**0**.2038	0.4331	0.2068

5.3. Multi-Label classification

Multi-label classification (MLC) is similar to MTR since both deal with the prediction of multiple variables using a common set of input variables. However, the targets in MLC are binary variables $\{-1, +1\}$ instead of continuous variables in MTR. We call the sample with label $+1$ the *positive* sample, and the one with label -1 the *negative* sample. The conventional approach of MLC is to learn a separate binary classification model for each task. Here we aim to evaluate different CCA models for multi-label classification. In [26], CCA methods were used to project the data into a lower-dimensional space in which the L_2-regularized L_1-loss support vector classification model [9] is applied for classifying each label separately.

Training stage

(1) Input training datasets $X_a \in \mathbb{R}^{n\times q}$ and $X_b \in \mathbb{R}^{m\times q}$.
(2) Centralize $X_a \leftarrow X_a - \mu_a \mathbf{1}_q^{\mathrm{T}}$ and $X_b \leftarrow X_b - \mu_b \mathbf{1}_q^{\mathrm{T}}$, where the centres $\mu_a = (1/q)X_a\mathbf{1}_q$ and $\mu_b = (1/q)X_b\mathbf{1}_q$.
(3) Apply the range constrained CCA, OCCA or pOCCA models to the dataset pair (X_a, X_b) and obtain transformation matrix pair (P_a, P_b) for a chosen positive integer k, $1 \le k \le \min\{m, n, q\}$. Note that for the pOCCA, orthogonal constraint is only imposed on X_a.
(4) Define the transformed data $Z = P_a^{\mathrm{T}} X_a$, which is most correlated to the target.
(5) Use the L_2-regularized L_1-loss support vector classification model [9] to learn a binary classification function $h(z) = \mathrm{sign}(w^{\mathrm{T}}z + b)$ from any input z to its output target $h(z) \in \{-1, 1\}$. Specifically, given the transformed data $Z = [z_1, \ldots, z_q]$ and its true target matrix X_b with the jth target of the ith data point denoted by $(X_b)_{j,i}$, solve the following optimization problem with respect to $W = [w_1, \ldots, w_m] \in \mathbb{R}^{n\times m}$ and $b = [b_1, \ldots, b_m] \in \mathbb{R}^m$:

$$\min_{W,b} \sum_{j=1}^{m} \left[\frac{1}{2} w_j^{\mathrm{T}} w_j + \gamma \sum_{i=1}^{q} \max\left\{0, 1 - (X_b)_{j,i}\,(w_j^{\mathrm{T}} z_i + b_j)\right\} \right], \tag{27}$$

where γ is the regularization parameter. The liblinear toolbox[5] is used to solve (27).

Testing and evaluation stage

(1) Input testing datasets $X'_a \in \mathbb{R}^{n\times q'}$ and $X'_b \in \mathbb{R}^{m\times q'}$, transformation matrix P_a, and centres μ_a and μ_b.
(2) Centralize testing data $X'_a \leftarrow X'_a - \mu_a \mathbf{1}_{q'}^{\mathrm{T}}$
(3) Apply the learned function h to predict the target value of the testing dataset X'_a, i.e. for $i = 1, \ldots, q'$ and $j = 1, \ldots, m$,

$$f_j(z'_i) = w_j^{\mathrm{T}} z'_i + b_j,$$
$$(X_*)_{j,i} = \mathrm{sign}(f_j(z'_i))$$

where $Z' = P_a^{\mathrm{T}} X'_a \equiv [z'_1, z'_2, \ldots, z'_{q'}]$.

(4) Given the true target matrix X_b, accuracy (ACC) and area under the curve (AUC) [4] are used for evaluating the learning performance:

$$\mathrm{Accuracy}(X_*, X_b) = \frac{1}{q'm} \sum_{i=1}^{q'} \sum_{j=1}^{m} \delta((X_*)_{j,i}, (X_b)_{j,i}),$$

$$\mathrm{AUC}(h(Z'), X'_b) = \frac{1}{m} \sum_{j=1}^{m} \frac{\sum_{i\in\mathcal{S}_+^j} \sum_{j\in\mathcal{S}_-^j} 1_{f_j(z'_i) > f_j(z'_j)}}{|\mathcal{S}_+^j| \times |\mathcal{S}_-^j|}$$

where $\delta((X_*)_{j,i}, (X_b)_{j,i}) = 1$ if $(X_*)_{j,i} = (X_b)_{j,i}$ and 0 otherwise, the set of positive samples $\mathcal{S}_+^j = \{i : (X_b)_{j,i} = +1, 1 \le i \le q'\}$ of the jth label, the set of negative samples $\mathcal{S}_-^j = \{i : (X_b)_{j,i} = -1, 1 \le i \le q'\}$ of the jth label, and $1_{a>b} = 1$ if the predicate $a > b$ is true, and 0 otherwise.

The datasets used in this numerical experiments are from Mulan: A Java Library for Multi-Label Learning[6] except rcv1subset_topics[7] and Bibtex.[8] The training and testing are provided for evaluating the classification performance. Both ACC and AUC scores are computed for each label and the averaged performance over all labels is reported. That ACC and AUC are closer to 1 means better performance.

Table 3 shows the statistics of the 10 datasets used in our experiments including data splitting for training and testing, the number of features, and the number of labels, as well as the classification scores ACC and AUC of the three methods on the 10 datasets by varying parameters k and γ (details will be shown later). The best results are shown in bold. In terms of ACC and AUC, OCCA model achieves best results on 7 out of the 10 datasets. pOCCA also shows quite good results, very close to these by OCCA and outperforms CCA in most cases. It implies that the orthogonal constraints that are imposed in OCCA work well for MLC, too.

We further conduct the sensitivity analysis of our proposed methods against CCA in terms of k and the parameter C of SVM varied in $\{0.01, 0.1, 1, 10, 100\}$. As discussed above, k is constrained to be no larger than $\min\{m, n, q\}$. Hence, we evaluate different k values according to the statistics of MLC datasets based on the following rules: for datasets with the number of labels $m < 20$, each k from 1 to m is evaluated; for $20 \le m < 200$, we choose

$$k \in \{1, 2, \ldots, 10\} \cup \{20, 40, 60, \ldots, m\} \cup \{m\};$$

otherwise, $k \in \{1, 2, \ldots, 10\} \cup \{30, 60, 90, \ldots, m\} \cup \{m\}$. Figure 1 shows the best ACC and AUC over all evaluated γs with respect to the varied number ks on four datasets. Figure 2 shows the best ACC and AUC over all evaluated k with respect to varying γ on the same four datasets. According to Figures 1 and 2, we have the following observations. (1) For all three methods, both ACC and AUC increase when k increases at the beginning and then reach to be either stable or degraded. So, the maximum reachable k is not always the best. (2) The varied regularization parameter γ in SVM can properly reflect the generalization performances of SVM classifier, where the peak is located differently for the four datasets and a properly chosen γ is necessary to achieve the best performance. (3) our methods

Table 3. ACC and AUC of MLC by range constrained traditional CCA (4), OCCA (11) and pOCCA (15).

Data	Training	Testing	Features	Labels	CCA		OCCA		pOCCA	
					ACC	AUC	ACC	AUC	ACC	AUC
Mediamill	30993	12914	120	101	0.6077	0.7032	**0.6259**	**0.7011**	0.5901	0.6995
Scene	1211	1196	294	6	0.6787	0.7636	**0.6968**	**0.7854**	0.6839	0.7724
Birds	322	323	260	19	0.6071	0.6279	**0.7792**	0.7236	0.7103	**0.7618**
Corel5k	4500	500	499	374	0.6387	0.4638	**0.8001**	**0.4842**	0.6305	0.4817
Emotions	391	202	72	6	**0.7368**	0.7613	0.7021	0.7272	0.7310	**0.7652**
Yeast	1500	917	103	14	**0.5920**	0.6284	0.5908	0.6242	0.5894	**0.6308**
rcv1_topics	3000	3000	47236	101	0.5191	0.5415	**0.6891**	**0.8092**	0.6035	0.7714
Bibtex_0	2515	4880	1836	159	0.5670	0.6873	**0.6591**	**0.7892**	0.5744	0.7431
Bibtex_1	2515	4880	1836	159	0.5799	0.6895	**0.6623**	**0.7860**	0.5892	0.7472
Delicious	12920	3185	500	983	**0.9384**	0.6842	0.8662	**0.6993**	0.8662	0.6956

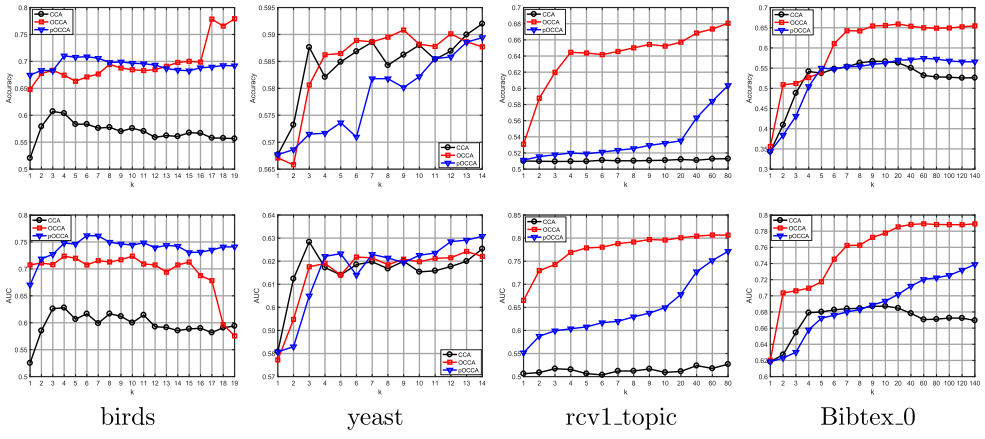

Figure 1. ACC and AUC of MLC by varying k on four datasets.

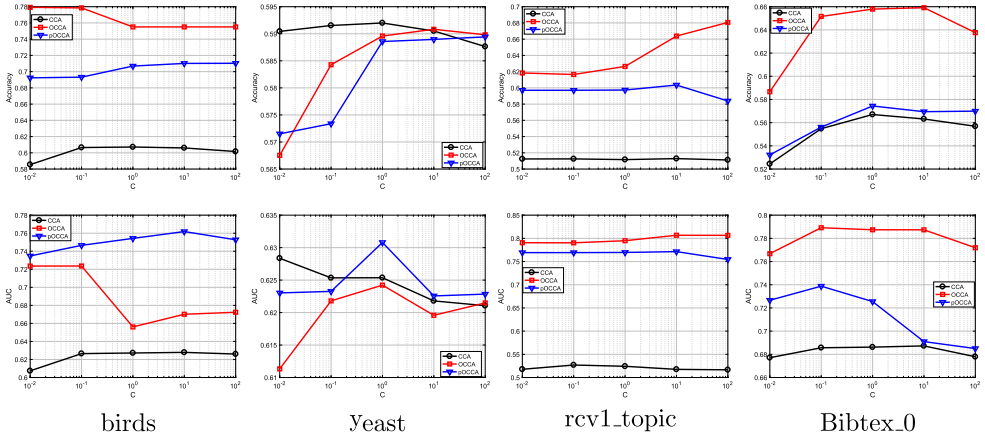

Figure 2. ACC and AUC of MLC by varying C of SVM on four datasets.

in general show significantly better results than CCA over various k and γ. Even for the dataset such as yeast, our methods still demonstrate very competitive results over all set of parameters.

6. Concluding remarks

We have proposed a range constrained OCCA model and a range constrained partial OCCA (pOCCA) model. It draws inspiration from the standard SVD, where singular vectors can be defined one-by-one. Upon imposing appropriate orthogonality conditions, the singular vectors so defined are provably the same as the ones otherwise defined all together.

We have tested both models and our implementations on various datasets from three data science applications. Our experimental results show that the proposed range constrained OCCA and pOCCA models outperforms the traditional CCA model in most of cases, and often the improvements are dramatic. Our pOCCA in multi-task regression and multi-label classification also show improved results, and especially on the datasets

where the correlation among labels can be very important as discussed in [6,14]. These empirical observations are consistent with our initial assumption that (partial) orthogonal constraints can be potentially used to boost the performance of the traditional CCA.

In this paper, our main intention is to show the effectiveness of the proposed models. Currently, they are implemented through full explicit SVD and that can be very expensive and memory intensive for high-dimensional problems. Our next stage is to exploit partial SVD via Lanczos type methods such as the Golub-Kahan bi-diagonalization [12] for large sized problems.

Ideally, a more direct orthogonal CCA model is to replace the constraints (3b) or (4b) by $P_a^{\mathrm{T}}P_a = P_b^{\mathrm{T}}P_b = I_k$ to yield the optimization problem

$$\max_{P_a\in\mathbb{R}^{n\times k},\, P_b\in\mathbb{R}^{m\times k}} \quad \frac{\mathrm{tr}(P_a^{\mathrm{T}}CP_b)}{\sqrt{\mathrm{tr}(P_a^{\mathrm{T}}AP_a)\mathrm{tr}(P_b^{\mathrm{T}}BP_b)}}, \tag{28a}$$

$$\text{subject to} \quad P_a^{\mathrm{T}}P_a = I_k, \quad P_b^{\mathrm{T}}P_b = I_k. \tag{28b}$$

Additional range constraints $\mathcal{R}(P_a) \subset \mathcal{R}(X_a)$ and $\mathcal{R}(P_b) \subset \mathcal{R}(X_b)$ can be included when desired. Unfortunately, there is no existing numerical linear algebra technique for us to readily build upon in order to solve (28) efficiently and robustly. Several generic optimization methods (for example, the line-search methods and the trust-region method) have been extended from the traditional Euclidean space case (e.g. [22,28]) to the Riemannian mannifold [1,2,8] and may be used to solve (28), as suggested in [5]. Also, the problem (28) is about maximization over the product of two Stiefel manifolds, and thus possibly some generic Stiefel manifold-based optimization methods [2, Section 9.4], [8], as well as some recent improvements [10,11,17,18,30] can be extended to deal with it. Nevertheless, a direct calling of a generic optimization approach to (28) may at best find local maximizers. In [32,33], it was numerically demonstrated that general optimization solvers on a Stiefel manifold for maximizing certain trace ratio-related function may converge slowly and yield less accurate solutions at convergence. Our development of rc-OCCA is a surrogate of (28) and both share a same solution only if $k = 1$. Our mantra in this paper has been to seek effective algorithms that can be robustly implemented by harvesting proven numerical linear algebra techniques, in this case SVD.

In the same spirit, a more straightforward pOCCA model than the one we proposed in Section 4 is

$$\max_{P_a\in\mathbb{R}^{n\times k},\, P_b\in\mathbb{R}^{m\times k}} \quad \frac{\mathrm{tr}(P_a^{\mathrm{T}}CP_b)}{\sqrt{\mathrm{tr}(P_a^{\mathrm{T}}AP_a)\mathrm{tr}(P_b^{\mathrm{T}}BP_b)}}, \tag{29a}$$

$$\text{subject to} \quad P_a^{\mathrm{T}}P_a = I_k, \quad P_b^{\mathrm{T}}BP_b = I_k. \tag{29b}$$

where P_a is forced to have orthonormal columns while P_b to have B-orthonormal columns. Again range constraints $\mathcal{R}(P_a) \subset \mathcal{R}(X_a)$ and $\mathcal{R}(P_b) \subset \mathcal{R}(X_b)$ can be included when desired. As for OCCA (28), it is not clear how to solve (29) for $k > 1$ with existing numerical linear algebra techniques, either. That leads us to design the range constrained pOCCA model the way in Section 4.

Notes

1. Private communications, 2019.
2. https://archive.ics.uci.edu/ml/datasets/Multiple+Features.
3. https://www.csie.ntu.edu.tw/ ∼ cjlin/liblinear/.
4. http://mulan.sourceforge.net/datasets-mtr.html.
5. https://www.csie.ntu.edu.tw/ ∼ cjlin/liblinear/.
6. http://mulan.sourceforge.net/datasets-mlc.html.
7. https://www.csie.ntu.edu.tw/ ∼ cjlin/libsvmtools/datasets/multilabel.html.
8. http://manikvarma.org/downloads/XC/XMLRepository.html.

Acknowledgements

The authors wish to thank anonymous referees for their constructive comments and suggestions that improved the presentation.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The research of Zhang was supported in part by the National Natural Science Foundations of China grant numbers NSFC-11671246 and NSFC-91730303. Bai was supported in part by NSF grants CCF-1527091 and DMS-1913364. Li was supported in part by NSF grants 1527104 and DMS-1719620.

Notes on contributors

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 received her PhD degree from Department of Mathematics, University of California, San Diego, USA, in 2014. Her research interests include large scale optimization, polynomial optimization and machine learning.

Lei-hong Zhang is a professor with the School of Mathematical Sciences, Soochow University, Suzhou, China. He received the PhD degree in mathematics from Hong Kong Baptist University, China in 2008. His research interest includes numerical optimization, eigenvalue problems and machine learning.

Zhaojun Bai is a professor in the Department of Computer Science and Department of Mathematics, University of California, Davis. He obtained his PhD from Fudan University, China in 1988. His main research interests include linear algebra algorithm design and analysis, mathematical software engineering and applications in computational science and engineering, and data science.

Ren-cang Li is a professor with the Department of Mathematics, University of Texas at Arlington, Texas, USA. He received his PhD degree in applied mathematics from the University of California at Berkeley in 1995. 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.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms On Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008.
[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (eds.), *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, PA, 2000.

[3] D. Chu, L. Liao, M.K. Ng, and X. Zhang, *Sparse canonical correlation analysis: new formulation and algorithm*, IEEE Trans. Pattern Anal. Mach. Intell. 35(12) (2013), pp. 3050–3065.
[4] C. Cortes and M. Mohri, *AUC optimization vs. error rate minimization*, in *Advances in Neural Information Processing Systems* 16, S. Thrun, L.K. Saul, and B. Scholkopf, eds., 2004, pp. 313–320.
[5] J.P. Cunningham and Z. Ghahramani, *Linear dimensionality reduction: survey, insights, and generalizations*, J. Mach. Learning Res. 16 (2015), pp. 2859–2900.
[6] K. Dembszynski, W. Waegeman, W. Cheng, and E. Hüllermeier, *On label dependence in multilabel classification*, LastCFP: ICML Workshop on Learning from Multi-label data, Ghent University, KERMIT, Department of Applied Mathematics, Biometrics and Process Control, 2010.
[7] S. Džeroski, D. Demšar, and J. Grbović, *Predicting chemical parameters of river water quality from bioindicator data*, Appl. Intell. 13(1) (2000), pp. 7–17.
[8] A. Edelman, T.A. Arias, and S.T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl. 20(2) (1999), pp. 303–353.
[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, *Liblinear: A library for large linear classification*, J. Mach. Learn. Res. 9(Aug) (2008), pp. 1871–1874.
[10] B. Gao, X. Liu, X.J. Chen, and Y. Yuan, *A new first-order algorithmic framework for optimization problems with orthogonality constraints*, SIAM J. Optim. 28(1) (2018), pp. 302–332.
[11] B. Gao, X. Liu, and Y. Yuan, *First-order algorithms for optimization problems with orthogonality constraints*, Oper. Res. Trans. 21(4) (2017), pp. 57–68.
[12] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, MD, 2013.
[13] D.R. Hardoon, S. Szedmak, and J. Shawe-Taylor, *Canonical correlation analysis: an overview with application to learning methods*, Neural Comput. 16(12) (2004), pp. 2639–2664.
[14] B. Hariharan, L. Zelnik-Manor, M. Varma, and S. Vishwanathan, *Large scale max-marginmultilabel classification with priors*, in *Proceedings of the 27th International Conference on Machine Learning*, J. Fürnkranz and T. Joachims, eds., Omnipress, Madison, WI, 2010, pp. 423–430.
[15] E. Hatzikos, G. Tsoumakas, G. Tzanisand, N. Bassiliades, and I. Vlahavas, *An empirical study on sea water quality prediction*, Knowl. Based Syst. 21(6) (2008), pp. 471–478.
[16] H. Hotelling, *Relations between two sets of variates*, Biometrika 28(3-4) (1936), pp. 321–377.
[17] J. Hu, A. Milzarek, Z. Wen, and Y. Yuan, *Adaptive quadratically regularized Newton method for Riemannian optimization*, SIAM J. Matrix Anal. Appl. 39 (2018), pp. 1181–1207.
[18] B. Jiang and Y.-H. Dai, *A framework of constraint preserving update schemes for optimization on stiefel manifold*, Math. Program. 153 (2015), pp. 535–575.
[19] A. Karalič and I. Bratko, *First order regression*, Mach. Learn. 26(2-3) (1997), pp. 147–176.
[20] L. Mackey, *Deflation methods for sparse PCA*, in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio and L. Bottou, eds., NIPS, 2008, pp. 1017–1024.
[21] R.J. Muirhead, *Aspects of Multivariate Statistical Theory*, 2nd ed., Wiley, New York, NY, 2005.
[22] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer, Berlin, 2006.
[23] G. Rong, C. Jin, S. Kakade, and P.N.A. Sidford, *Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis*, in *Proceedings of the 33rd International Conference on Machine Learning*, M. Balcan and K. Weinberger, eds., JMLR, 2016, pp. 2741–2750.
[24] 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*, IEEE, Istanbul, 2013, pp. 151–157.
[25] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas, *Multi-target regression via input space expansion: treating targets as inputs*, Mach. Learn. 104(1) (2016), pp. 55–98.
[26] L. Sun, S. Ji, and J. Ye, *A least squares formulation for canonical correlation analysis*, in *Proceeding of the 25th International Conference on Machine learning*, A. McCallum and S. Roweis eds., ACM, New York, 2008, pp. 1024–1031.

[27] Q.-S. Sun, S.-G. Zeng, Y. Liu, P.-A. Heng, and D.-S. Xia, *A new method of feature fusion and its application in image recognition*, Pattern. Recognit. 38(12) (2005), pp. 2437–2448.
[28] W. Sun and Y. Yuan, *Optimization Theory and Methods*, Springer, Berlin, 2006.
[29] A. Tsanas and A. Xifara, *Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools*, Energy Build. 49 (2012), pp. 560–567.
[30] Z. Wen and W. Yin, *A feasible method for optimization with orthogonality constraints*, Math. Program. 142(1-2) (2013), pp. 397–434.
[31] I-C. Yeh, *Modeling slump flow of concrete using second-order regressions and artificial neural networks*, Cem. Concr. Compos. 29(6) (2007), pp. 474–480.
[32] L.-H. Zhang and R.-C. Li, *Maximization of the sum of the trace ratio on the Stiefel manifold, I: theory*, Sci. China Math. 57(12) (2014), pp. 2495–2508.
[33] L.-H. Zhang and R.-C. Li, *Maximization of the sum of the trace ratio on the Stiefel manifold, II: computation*, Sci. China Math. 58(7) (2015), pp. 1549–1566.