Applied Mathematics and Computation 383 (2020) 125272

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

tSSNALM: A fast two-stage semi-smooth Newton augmented Lagrangian method for sparse CCA

Xianchao Xiu[a], Ying Yang[a,*], Lingchen Kong[b], Wanquan Liu[c]

[a] *Department of Mechanics and Engineering Science, Peking University, Beijing, China*
[b] *Department of Applied Mathematics, Beijing Jiaotong University, Beijing, China*
[c] *Department of Computing, Curtin University, Perth, WA Australia*

ARTICLE INFO

Article history:
Received 17 May 2019
Revised 7 February 2020
Accepted 29 March 2020

2010 MSC:
62
90

Keywords:
Sparse canonical correlation analysis
Semi-smooth Newton
Augmented Lagrangian method
Duality

ABSTRACT

Canonical correlation analysis (CCA) is a very useful tool for measuring the linear relationship between two multidimensional variables. However, it often fails to extract meaningful features in high-dimensional settings. This motivates the sparse CCA problem, in which ℓ_1 constraints are applied to the canonical vectors. Although some sparse CCA solvers exist in the literature, we found that none of them is efficient. We propose a fast two-stage semi-smooth Newton augmented Lagrangian method (tSSNALM) to solve sparse CCA problems, and we provide convergence analysis. Numerical comparisons between our approach and a number of state-of-the-art solvers, on simulated data sets, are presented to demonstrate its efficiency. To the best of our knowledge, this is the first time that duality has been integrated with a semi-smooth Newton method for solving sparse CCA.

1. Introduction

Canonical correlation analysis (CCA) was originally proposed in Hotelling [1], and is now one of the most classical and important tools in multivariate data analysis for finding the correlation between two sets of multidimensional variables. We refer to [2–8] for broad applications in areas such as economics, geography, medicine, chemistry, signal processing, fault detection, and deep learning. More references can be found in a recent survey paper [9].

1.1. Overview of the Problem

For two data matrices $X \in \mathbb{R}^{n\times p}$ $(n \ll p)$ and $Y \in \mathbb{R}^{n\times q}$ $(n \ll q)$, CCA seeks a pair of linear transformations $\beta \in \mathbb{R}^p$ and $\theta \in \mathbb{R}^q$ such that the projected variables $X\beta$ and $Y\theta$ in the lower-dimensional space are maximally correlated. That is, it aims to solve the optimization problem

$$\begin{aligned} &\max_{\beta,\theta}\ \operatorname{Cov}(X\beta, Y\theta) \\ &\ \text{s.t.}\ \ \operatorname{Var}(X\beta) = 1,\ \ \operatorname{Var}(Y\theta) = 1, \end{aligned}$$

[*] Corresponding Author.
E-mail addresses: xcxiu@bjtu.edu.cn (X. Xiu), yy@pku.edu.cn (Y. Yang), lchkong@bjtu.edu.cn (L. Kong), W.Liu@curtin.edu.au (W. Liu).

https://doi.org/10.1016/j.amc.2020.125272
0096-3003/© 2020 Elsevier Inc. All rights reserved.

where $\mathrm{Cov}(X\beta, Y\theta)$ is the covariance, defined as

$$\mathrm{Cov}(X\beta, Y\theta) = \frac{1}{n}\sum_{i=1}^{n}(X_i^T\beta)(Y_i^T\theta).$$

We assume that their expectations are zero. Thus, the CCA problem is equivalent to

$$\begin{aligned}&\min_{\beta,\theta}\ -\beta^T X^T Y\theta\\ &\ \text{s.t.}\ \ \|X\beta\|^2 = 1,\ \ \|Y\theta\|^2 = 1.\end{aligned} \tag{1}$$

As noted in Martin et al. [10], the classical CCA model (1) is easy to solve because it is equivalent to obtaining the singular decomposition of X^TY. However, it often fails to extract meaningful features in high-dimensional settings, where the dimensions p and q could be much larger than the sample size n. This is because meaningless solutions exist with correlations equal to one [11,12]. In addition, a high portion of features are not informative in data analysis. For example, when the dimension is proportional to the sample size such that $\lim_{n\to\infty} p/n = \gamma \in (0, 1)$, the largest eigenvalue is less than $\sqrt{\gamma}$, and the leading sample principal eigenvector could be asymptotically orthogonal to the leading population principal eigenvector almost surely, see [13]. To deal with these challenges, Witten et al. [14] developed sparse canonical correlation analysis (SCCA) by imposing ℓ_1-constraints on β and θ:

$$\begin{aligned}&\min_{\beta,\theta}\ -\beta^T X^T Y\theta + \lambda\|\beta\|_1 + \mu\|\theta\|_1\\ &\ \text{s.t.}\ \ \|X\beta\|^2 = 1,\ \ \|Y\theta\|^2 = 1,\end{aligned} \tag{2}$$

where $\|\cdot\|_1$ is the sum of absolute values of all entries, and λ, $\mu > 0$ are two weighting parameters to control the sparsity of variables β, θ.

Note that the objective of SCCA model (2) is biconvex, meaning if we fix β, the resulting objective is convex respect to θ, and if we fix θ, the objective is convex respect to β. However, the constraints $\|X\beta\|^2 = 1,\ \|Y\theta\|^2 = 1$ are not convex, which results in a very challenging optimization problem. This, together with the above-mentioned applications, has inspired many researchers to study how to solve CCA efficiently. By substituting the identity matrix I for X^TX and Y^TY, they dropped X and Y from $\|X\beta\|^2 = 1$ and $\|Y\theta\|^2 = 1$, and obtained

$$\begin{aligned}&\min_{\beta,\theta}\ -\beta^T X^T Y\theta + \lambda\|\beta\|_1 + \mu\|\theta\|_1\\ &\ \text{s.t.}\ \ \|\beta\|^2 = 1,\ \ \|\theta\|^2 = 1.\end{aligned}$$

They then relaxed the non-convex constraints $\|\beta\|^2 = 1$ and $\|\theta\|^2 = 1$ to convex surrogates to obtain the problem

$$\begin{aligned}&\min_{\beta,\theta}\ -\beta^T X^T Y\theta + \lambda\|\beta\|_1 + \mu\|\theta\|_1\\ &\ \text{s.t.}\ \ \|\beta\|^2 \le 1,\ \ \|\theta\|^2 \le 1.\end{aligned}$$

For their algorithm, they designed an efficient toolbox PMA (written in R). Although PMA is easy to use, the assumption is not very realistic in high-dimensional settings. As a result, Gao et al. [15] proposed to relax $\|X\beta\|^2 = 1$ and $\|Y\theta\|^2 = 1$, and solved the problem

$$\begin{aligned}&\min_{\beta,\theta}\ -\beta^T X^T Y\theta + \lambda\|\beta\|_1 + \mu\|\theta\|_1\\ &\ \text{s.t.}\ \ \|X\beta\|^2 \le 1,\ \ \|Y\theta\|^2 \le 1.\end{aligned} \tag{3}$$

The reasoning is that solutions are often on the boundary, and the convex hull is the best relaxation. They discussed the problem's statistical properties, and established an adaptive estimation procedure with computational lower bound. This method, called CoLaR, includes two stages. In the first stage, matrix lifting is used to solve (3). In the second stage, the solution is refined by adopting the group Lasso.

Recently, Suo et al. [16,17] proposed an alternating minimization algorithm (AMA) for solving the SCCA problem (3). AMA solves two subproblems each iteration, and each subproblem is solved by a linearized alternating direction method of multipliers algorithm.

Although numerical experiments have demonstrated effectiveness, we must point out that none of these algorithms has a convergence analysis, and cannot ensure high efficiency for solving large-scale SCCA.

1.2. Contributions

We aim to design a highly efficient optimization algorithm to solve the SCCA model (3) with convergence analysis. Since a two-stage technique, semi-smooth Newton method and augmented Lagrangian method are employed, it is natural for us to call our algorithm a two-stage semi-smooth Newton augmented Lagrangian method (tSSNALM). Compared to previous work, the main innovations of our proposed algorithm are as follows:

- We are the first to solve the dual formulation (7) rather than (5) itself. This is very important in high-dimensional problems. For example, X^TX and Y^TY are of size $p \times p$ and $q \times q$. For the dual optimization problem, we only need XX^T and YY^T of size $n \times n$. Thus, the cost is reduced from $O(p^3)$ and $O(q^3)$ to $O(n^3)(n \ll \max(p, q))$.
- We develop a fast two-stage semi-smooth Newton augmented Lagrangian method (tSSNALM) to optimize SCCA model (3), and the resulting subproblems either have closed-form solutions or can be solved by fast solvers. In addition, we prove that the sequence generated by tSSNALM converges to a local optimum.
- We conduct a variety of simulated examples to demonstrate that our proposed tSSNALM can achieve better performance than the existing state-of-the-art solvers CoLaR [15] and AMA [16].

In the next section, we introduce some definitions, review the augmented Lagrangian method, and present preliminary results on proximal mapping. In section 3, we propose a two-stage semi-smooth Newton augmented Lagrangian method to solve SCCA and analyze its convergence. In section 4, we conduct extensive numerical experiments to evaluate the performance of our proposed method in solving SCCA.

2. Preliminaries

We define some notations, briefly review the augmented Lagrangian method, and present some basic ideas of proximal mapping.

2.1. Notation

We use $\mathbb{R}^n$ and $\mathbb{R}^{n\times p}$ to denote the set of n-dimensional vectors and $n \times p$ matrices. For a vector $x \in \mathbb{R}^n$, let x_i denote its i-th entry. For two vectors x and y of the same size, we denote their inner product by $\langle x, y\rangle = \sum_{i=1}^n x_i y_i$. The corresponding ℓ_1-norm, ℓ_2-norm and ℓ_∞-norm are defined as

$$\|x\|_1 := \sum_{i=1}^n |x_i|, \;\; \|x\|_2 := \sqrt{\langle x, x\rangle}, \;\; \|x\|_\infty := \max\{|x_i|, i = 1, 2, \dots, n\}.$$

For simplicity, we always use $\|x\|$ to denote $\|x\|_2$. From the definition of sub-differential [18], we can derive that

$$\partial\|x\|_1 := \{\xi \mid \xi_i \in \mathrm{sign}(x_i)\},$$

where the sign operator is given by

$$\mathrm{sign}(x_i) := \begin{cases} 1 & \text{if } x_i > 0, \\ [-1, 1] & \text{if } x_i = 0, \\ -1 & \text{if } x_i < 0. \end{cases}$$

The closed ℓ_∞-ball with radius $\lambda \geq 0$ is defined as

$$B_{\lambda\infty}(x) := \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq \lambda\}.$$

For a nonempty convex set $\Omega \subset \mathbb{R}^n$, the indicator function is

$$\delta_\Omega(x) := \begin{cases} 0 & \text{if } x \in \Omega, \\ \infty & \text{otherwise.} \end{cases}$$

Finally, the distance from x to Ω is denoted by

$$\mathrm{dist}(x; \Omega) := \inf\{\|x - \omega\| \mid \omega \in \Omega\}.$$

2.2. Augmented Lagrangian Method (ALM)

The augmented Lagrangian method (ALM) is widely used in computer vision, image processing, statistical learning; see Boyd et al. [19] for a comprehensive review. Let us consider the convex composite problem

$$\begin{aligned} \min_{x,y} \quad & f(x) + g(y) \\ \text{s.t.} \quad & \mathcal{A}(x) + \mathcal{B}(y) = c, \end{aligned} \tag{4}$$

where $x, y \in \mathbb{R}^n$ are two variables, $\mathcal{A}, \mathcal{B} : \mathbb{R}^n \to \mathbb{R}^n$ are linear maps, and $c \in \mathbb{R}^n$ is a given vector. The augmented Lagrangian for (4) is

$$\mathcal{L}_\sigma(x, y; z) := f(x) + g(y) - \langle z, \mathcal{A}(x) + \mathcal{B}(y) - c\rangle + \frac{\sigma}{2}\|\mathcal{A}(x) + \mathcal{B}(y) - c\|^2,$$

where $\sigma > 0$ is a penalty parameter, and $z \in \mathbb{R}^n$ is the Lagrange multiplier. The augmented Lagrangian method (ALM) was originally proposed in Hestenes [20] and Powell [21]. The iterative scheme for solving problem (4) can be summarized as

$$\begin{cases} (x^{k+1}, y^{k+1}) = \arg\min_{x,y} \left\{ \mathcal{L}_\sigma(x, y; z^k) \right\}, \\ z^{k+1} = z^k - \tau\sigma_k\big(\mathcal{A}(x^{k+1}) + \mathcal{B}(y^{k+1}) - c\big), \\ \sigma_k \uparrow \sigma_\infty \le +\infty, \end{cases}$$

where $\tau \in \left(0, \frac{1+\sqrt{5}}{2}\right)$. In most cases, the resulting subproblems admit closed-form solutions, which makes the ALM particularly efficient. The next section presents some proximal mappings, which are exactly the corresponding solutions for subproblems in solving (3).

2.3. Proximal Mapping

Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a proper closed convex function. The Moreau envelope function $\phi_f(\cdot)$ with $\sigma > 0$ is defined as

$$\phi_{f/\sigma}(x) := \min_y \left\{ f(y) + \frac{\sigma}{2}\|y - x\|^2 \right\},$$

and the proximal mapping is the corresponding solution of the above optimization problem:

$$\mathrm{Prox}_{f/\sigma}(x) := \arg\min_y \left\{ f(y) + \frac{\sigma}{2}\|y - x\|^2 \right\}.$$

More illustrations can be found in [22,23]. Next, we recall some proximal mappings that we use in the algorithm analysis.

Example 2.1. If $f(y) = \|y\|_1$, the optimization problem

$$\min_y \left\{ \|y\|_1 + \frac{\sigma}{2}\|y - x\|^2 \right\}$$

has a unique optimal solution given by

$$\mathrm{Prox}_{f/\sigma}(x) = \mathtt{shrink}(x, 1/\sigma) := \mathrm{sign}(x) \cdot \max\{|x| - 1/\sigma, 0\}.$$

It is called the soft-thresholding operator; see [24] for more details.

Example 2.2. If $f(y) = \delta_{\lambda B_\infty}(y)$, the optimization problem

$$\min_y \left\{ \delta_{\lambda B_\infty}(y) + \frac{\sigma}{2}\|y - x\|_2^2 \right\}$$

has a unique optimal solution, i.e.,

$$\mathrm{Prox}_{\delta_{\lambda B_\infty}}(x) = \Pi_{\lambda B_\infty}(x) := x - \mathrm{sign}(x) \cdot \max\{|x| - \lambda, 0\}.$$

We end this section with the following proposition, which is called the Moreau identity [25].

Proposition 2.3. *Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a proper closed convex function, and f^* be its conjugate function defined in [18]. Then we have*

$$\mathrm{Prox}_{\sigma f}(x) + \sigma\mathrm{Prox}_{f^*/\sigma}(x/\sigma) = x,$$

where $\sigma > 0$ is a given parameter.

Indeed, this proposition illustrates the connection between the primal function and its conjugate function.

3. Optimization Algorithm and Convergence Analysis

In this section, we first describe the optimization algorithm in detail, then discuss the convergence analysis in section 3.4.

3.1. A Two-Stage Method for (3)

Although problem (3) is non-convex, it is biconvex. This motivates us to separate the problem into two stages:

- Fix θ, optimization problem (3) becomes

$$\begin{aligned} \min_\beta \quad & -\beta^T X^T Y\theta + \lambda\|\beta\|_1 \\ \text{s.t.} \quad & \|X\beta\|^2 \le 1. \end{aligned} \tag{5}$$

Algorithm 1 A Two-Stage Method for (3)

```
Input: $X \in \mathbb{R}^{n\times p}$, $Y \in \mathbb{R}^{n\times q}$, $\lambda, \mu > 0$, and $\theta^0 \in \mathbb{R}^q$.
while not converged do
    Fix $\theta^k$, compute $\beta^{k+1}$;
    Fix $\beta^{k+1}$, compute $\theta^{k+1}$;
end while
Output: $\beta \in \mathbb{R}^p$, $\theta \in \mathbb{R}^q$.
```

- Fix β, optimization problem (3) becomes

$$\begin{aligned} \min_{\theta} \quad & -\beta^T X^T Y\theta + \mu\|\theta\|_1 \\ \text{s.t.} \quad & \|Y\theta\|^2 \le 1. \end{aligned} \tag{6}$$

Thus, the algorithm can be summarized in Algorithm 1. The order of the updates can be reversed, i.e., it is possible first to optimize θ, followed by an optimization of β. In addition, there are several ways to define the stopping criterion of Algorithm 1. For example, one can consider the differences $\|\beta^{k+1} - \beta^k\|$ and $\|\theta^{k+1} - \theta^k\|$, or the primal and dual infeasibilities. The stopping criterion may also depend on the special structure of the given biconvex objective function.

3.2. An Augmented Lagrangian Method for (5)

We now show how to solve the minimization problem (5), and similarly (6). The dual optimization problem of (5) is

$$\begin{aligned} \min_{\alpha} \quad & \|\alpha - Y\theta\|^2 \\ \text{s.t.} \quad & \|X^T\alpha\|_\infty \le \lambda. \end{aligned} \tag{7}$$

Compared with (5), the variable α is only n-dimensional, where n is less than p. It is worth mentioning that this could be solved by an existing optimization solver, such as the quadratic programming solver SQOPT [26]. To make use of its structure, we reformulate (7) as an equivalent regularized problem

$$\begin{aligned} \min_{\alpha,\gamma} \quad & \frac{1}{2}\|\alpha - Y\theta\|^2 + \delta_{\lambda B_\infty}(\gamma) \\ \text{s.t.} \quad & X^T\alpha - \gamma = 0, \end{aligned} \tag{8}$$

where the auxiliary variable $\gamma \in \mathbb{R}^p$ liberates $X^T\alpha$ from $\|\cdot\|_\infty$. The augmented Lagrangian for (8) is then defined as

$$\mathcal{L}_\sigma(\alpha,\gamma;\beta) := \frac{1}{2}\|\alpha - Y\theta\|^2 + \delta_{\lambda B_\infty}(\gamma) - \langle\beta, X^T\alpha - \gamma\rangle + \frac{\sigma}{2}\|X^T\alpha - \gamma\|^2,$$

where $\beta \in \mathbb{R}^p$ is the Lagrange multiplier. Under the framework of ALM, the iterative scheme can be described in Algorithm 2.

Algorithm 2 An Augmented Lagrangian Method for (8)

```
Input: $X \in \mathbb{R}^{n\times p}$, $Y \in \mathbb{R}^{n\times q}$, $\theta \in \mathbb{R}^q$, $\tau \in (0, \frac{1+\sqrt{5}}{2})$, $\lambda > 0$, and $\beta^0 \in \mathbb{R}^p$.
while not converged do
    Compute $(\alpha^{k+1}, \gamma^{k+1}) \approx \arg\min_{\alpha,\gamma}\{\mathcal{L}_{\sigma_k}(\alpha,\gamma;\beta^k)\}$;
    Compute $\beta^{k+1} = \beta^k - \tau\sigma_k(X^T\alpha^{k+1} - \gamma^{k+1})$,and update $\sigma_k \to \sigma_\infty \le +\infty$;
end while
Output: $\alpha \in \mathbb{R}^n$, $\gamma \in \mathbb{R}^p$.
```

For the first subproblem, the solution may be computed inexactly. We refer to [27] and references therein for an algorithm and the global and local superlinear convergence analysis.

The main computational load is solving the first subproblem:

$$\min{}_{\alpha,\gamma} \frac{1}{2}\|\alpha - Y\theta\|^2 + \delta_{\lambda B_\infty}(\gamma) - \langle\beta^k, X^T\alpha - \gamma\rangle + \frac{\sigma}{2}\|X^T\alpha - \gamma\|^2. \tag{9}$$

We show how to solve (9) efficiently by using a semi-smooth Newton method.

3.3. A Semi-smooth Newton Method for (9)

Since (9) is strongly convex in α and γ, there exists a unique optimal solution (α^*, γ^*). For any α, we have

$$\varphi_k(\alpha) := \min_{\gamma} \mathcal{L}_{\sigma_k}(\alpha,\gamma;\beta^k)$$

$$
\begin{aligned}
&= \frac{1}{2}\|\alpha - Y\theta\|^2 - \frac{\|\beta^k\|^2}{2\sigma_k} + \min_{\gamma}\left\{\delta_{\lambda B_\infty}(\gamma) + \frac{\sigma_k}{2}\|X^T\alpha - \gamma - \beta^k/\sigma_k\|^2\right\}\\
&= \frac{1}{2}\|\alpha - Y\theta\|^2 - \frac{\|\beta^k\|^2}{2\sigma_k} + \frac{\sigma_k}{2}\|\mathrm{Prox}_{\lambda\|\cdot\|_1}(X^T\alpha - \beta^k/\sigma_k)\|^2,
\end{aligned}
$$

where the second equality follows from Moreau identity. Hence,

$$
\min_{\alpha} \ \varphi_k(\alpha) = \min_{\alpha,\gamma} \ \mathcal{L}_{\sigma_k}(\alpha,\gamma;\beta^k) = \min_{\alpha}\min_{\gamma} \ \mathcal{L}_{\sigma_k}(\alpha,\gamma;\beta^k). \tag{10}
$$

Denote

$$
\begin{aligned}
\gamma(\alpha) &:= \arg\min_{\gamma}\left\{\delta_{\lambda B_\infty}(\gamma) + \frac{\sigma_k}{2}\|X^T\alpha - \gamma - \beta^k/\sigma_k\|^2\right\}\\
&= \Pi_{\lambda B_\infty}(X^T\alpha - \beta^k/\sigma_k),
\end{aligned}
$$

which comes from Example 2.2. If $\min_\alpha \varphi_k(\alpha)$ has an optimal solution α^*, then (9) must have an optimal solution (α^*, $\gamma(\alpha^*)$).

Next, we analyze the properties of $\varphi_k(\alpha)$. Indeed, $\varphi_k(\alpha)$ is strongly convex, differentiable and strongly semi-smooth. We have

$$
\begin{aligned}
\nabla\varphi_k(\alpha) &= \alpha - Y\theta + \sigma_k X\mathrm{Prox}_{\lambda\|\cdot\|_1}(X^T\alpha - \beta^k/\sigma_k)\\
&= \alpha - Y\theta + X\mathrm{Prox}_{\sigma_k\lambda\|\cdot\|_1}(\sigma_k X^T\alpha - \beta^k),
\end{aligned}
$$

and

$$
\begin{aligned}
\partial(\nabla\varphi_k(\alpha)) &= I_n + X\partial\mathrm{Prox}_{\sigma_k\lambda\|\cdot\|_1}(\sigma_k X^T\alpha - \beta^k)(\sigma_k X^T)\\
&= I_n + \sigma_k X(\partial\mathrm{Prox}_{\sigma_k\lambda\|\cdot\|_1}(\sigma_k X^T\alpha - \beta^k)X^T.
\end{aligned}
$$

For a semi-smooth Newton method to be considered, we begin with discussing the search direction d:

$$
Vd = -\nabla\varphi_k(\alpha), \ V \in \partial(\nabla\varphi_k(\alpha)).
$$

Note that

$$
\partial(\nabla\varphi_k(\alpha)) = \left\{V \in \mathbb{R}^{n\times n} \mid V = I_n + \sigma_k XUX^T\right\},
$$

where

$$
U \in \partial\mathrm{Prox}_{\sigma_k\lambda\|\cdot\|_1}(\sigma_k X^T\alpha - \beta^k),
$$

$$
\partial\mathrm{Prox}_{\sigma_k\lambda\|\cdot\|_1}(w) = \left\{U = \mathrm{Diag}(u) \in \mathbb{R}^{p\times p} \mid u_i \in \left\{\begin{array}{ll} 1 & \text{if } |w_i| > \sigma_k\lambda,\\ [0,1] & \text{if } |w_i| = \sigma_k\lambda, \quad i = 1,\dots,n.\\ 0 & \text{if } |w_i| < \sigma_k\lambda. \end{array}\right.\right\}.
$$

We consider

$$
XU = [X_1,\dots,X_p]\begin{bmatrix} u_1 & & \\ & \ddots & \\ & & u_n \end{bmatrix} = [u_1X_1,\dots,u_pX_p].
$$

The computational cost of matrix multiplication associated with XUX^T is $O(n^2p)$. Hence, the cost of $V^{-1} = (I_n + \sigma_k XUX^T)^{-1}$ is $O(n^2p) + O(n^3)$.

These computational costs are too great when the dimensions of X are large and can make commonly employed approaches such as Cholesky factorization and the conjugate gradient method inappropriate for computing V^{-1}. Further, if X is sparse, the computational costs can be reduced substantially. We show how this can be done by taking full advantage of the sparsity of X.

For convenience of description, we always choose

$$
u_i = \left\{\begin{array}{ll} 1 & \text{if } |u_i| \geq \sigma_k\lambda,\\ 0 & \text{if } |u_i| < \sigma_k\lambda. \end{array}\right.
$$

We then have $U^2 = U, UU^T = U$. Denoting

$$
J := \{i \in \{1,2,\dots,n\} \mid u_i = 1\}
$$

and $r := |J| \leq n$, we obtain

$$
XUX^T = (XU)(XU)^T = X_JX_J^T.
$$

Now the computational cost of $X_JX_J^T$ is only $O(n^2r)$. When $r \ll n(\ll p)$, $O(n^2r)$ is much smaller than $O(n^2p)$. Moreover, we can make use of the Sherman-Morrison-Woodbury formula [28] to get the inverse of V by inverting a much smaller $r \times r$ matrix as follows:

$$V^{-1} = (I_n + \sigma_k X_J X_J^T)^{-1} = I_n - X_J(I_r/\sigma_k + X_J^T X_J)^{-1} X_J^T.$$

Overall, by considering the sparsity, the computational cost of solving the direction can be reduced from $O(n^2p) + O(n^3)$ to $O(n^2r) + O(nr^2) + O(r^3)$, which is a great improvement in high-dimensional settings.

Next, we discuss the stopping criterion. Denoting

$$\Phi_k(\alpha, \gamma) := \mathcal{L}_{\sigma_k}(\alpha, \gamma; \beta^k), \tag{11}$$

we have

$$\Phi_k(\alpha^{k+1}, \gamma^{k+1}) - \inf \Phi_k(\alpha, \gamma) = \varphi_k(\alpha^{k+1}) - \varphi_k(\alpha^*).$$

Because φ_k is convex, we see that

$$-(\varphi_k(\alpha^*) - \varphi_k(\alpha^{k+1})) \le \nabla\varphi_k(\alpha^{k+1})^T(-\alpha^* + y^{k+1}) \le \|\nabla\varphi_k(\alpha^{k+1})\| \|\alpha^{k+1} - \alpha^*\|.$$

Moreover, as φ_k is strongly convex with modulus at least γ, we derive

$$\varphi_k(\alpha^*) - \varphi_k(\alpha^{k+1}) \ge \nabla\varphi_k(\alpha^{k+1})^T(\alpha^* - \alpha^{k+1}) + \frac{1}{2}\|\alpha^* - \alpha^{k+1}\|^2$$

and

$$\varphi_k(\alpha^{k+1}) - \varphi_k(\alpha^*) \ge \nabla\varphi_k(\alpha^*)^T(\alpha^{k+1} - \alpha^*) + \frac{1}{2}\|\alpha^{k+1} - \alpha^*\|^2.$$

Using the fact that $\nabla\varphi_k(\alpha^*) = 0$, we have

$$-\nabla\varphi_k(\alpha^{k+1})^T(\alpha^* - \alpha^{k+1}) \ge \|\alpha^* - \alpha^{k+1}\|^2.$$

From the Cauchy-Schwarz inequality, it is easy to obtain

$$\|\nabla\varphi_k(\alpha^{k+1})\| \ge \|\alpha^* - \alpha^{k+1}\|,$$

and then

$$\Phi_k(\alpha^{k+1}, \gamma^{k+1}) - \inf \Phi_k(\alpha, \gamma) = \varphi_k(\alpha^{k+1}) - \varphi_k(\alpha^*) \le \|\nabla\varphi_k(\alpha^{k+1})\|^2.$$

On the other hand,

$$\begin{aligned}\partial\Phi_k(\alpha^{k+1}, \gamma^{k+1}) &= \begin{pmatrix} \alpha^{k+1} - Y\theta - X\beta^k + \sigma_k X(X^T\alpha^{k+1} - \gamma^{k+1}) \\ \partial\delta_{\lambda B_\infty}(\gamma^{k+1}) + \beta^k + \sigma_k(X^T\alpha^{k+1} - \gamma^{k+1}) \end{pmatrix} \\ &= \begin{pmatrix} y^{k+1} + b - Ax^{k+1} \\ \partial\delta_{\lambda B_\infty}(\gamma^{k+1}) - \beta^{k+1} \end{pmatrix}.\end{aligned}$$

The differential of $\varphi_k(\alpha^{k+1})$ with respect to α is given as

$$\begin{aligned}\nabla\varphi_k(\alpha^{k+1}) &= \alpha^{k+1} - Y\theta + \sigma_k X \mathrm{Prox}_{\lambda\|\cdot\|_1}(X^T\alpha^{k+1} - \beta^k/\sigma_k) \\ &= \alpha^{k+1} - Y\theta + X\beta^{k+1}.\end{aligned}$$

The differential of (11) with respect to γ is

$$\begin{aligned}&0 \in \partial\delta_{\lambda B_\infty}(\gamma^{k+1}) + \sigma_k\gamma^{k+1} - (\sigma_k X^T\alpha^{k+1} - \beta^k) \\ \Leftrightarrow\ &0 \in \partial\delta_{\lambda B_\infty}(\gamma^{k+1}) - \beta^{k+1}.\end{aligned}$$

From the above three relations, we have

$$\begin{pmatrix} \nabla\varphi_k(\alpha^{k+1}) \\ 0 \end{pmatrix} \in \partial\Phi_k(\alpha^{k+1}, \gamma^{k+1}).$$

Thus, $\mathrm{dist}(0, \partial\Phi_k(\alpha^{k+1}, \gamma^{k+1})) = \|\nabla\varphi_k(\alpha^{k+1})\|$.

According to [18], we can finally define the stopping criterion as follows:

$$\begin{cases} (\mathrm{A}) \quad \Phi_k(\alpha^{k+1}, \gamma^{k+1}) - \inf \Phi_k(\alpha, \gamma) \le \frac{\varepsilon_k^2}{2\sigma_k}, \\ (\mathrm{B1}) \ \ \Phi_k(\alpha^{k+1}, \gamma^{k+1}) - \inf \Phi_k(\alpha, \gamma) \le \frac{\delta_k^2}{2\sigma_k}\|\beta^{k+1} - \beta^k\|^2, \\ (\mathrm{B2}) \ \ \mathrm{dist}(0, \partial\Phi_k(\alpha^{k+1}, \gamma^{k+1})) \le \frac{\delta_k'}{2\sigma_k}\|\beta^{k+1} - \beta^k\|, \end{cases}$$

where $\sum\limits_{k=0}^{\infty} \varepsilon_k < \infty$, $\sum\limits_{k=0}^{\infty} \delta_k < \infty$, and $\delta_k' \to 0$. Overall, all these together guarantee the global convergence.

Algorithm 3 A Semi-smooth Newton Method for (9)

```
Input: $X \in \mathbb{R}^{n\times p}$, $Y \in \mathbb{R}^{n\times q}$, $\theta \in \mathbb{R}^{q}$, $\lambda > 0$, and $\beta^k \in \mathbb{R}^{p}$.
while not converged do
    Set $U \in \partial \mathrm{Prox}_{\lambda\|\cdot\|_1}(X^T\alpha^{(j)} - \beta^k/\sigma_k)$ and $V = I_n + \sigma_k XUX^T$. Solve the linear system
        $V^j d = -\nabla\varphi_k(\alpha^{(j)})$;
    Compute $\alpha_j = \delta^{m_j}$, where $m_j$ is the first nonnegative integer $m$ for which
        $\varphi_k(\alpha^{(j)} + \delta^m d^j) \le \varphi_k(\alpha^{(j)}) + \mu\delta^m(\nabla\varphi_k(\alpha^{(j)}))^T d^j$;
    Compute $\alpha^{(j+1)} = \alpha^{(j)} + \alpha_j d^j$;
end while
Output: $\alpha \in \mathbb{R}^n, \gamma \in \mathbb{R}^p$.
```

Finally, we present our proposed semi-smooth Newton method for solving (9) in Algorithm 3. Compared with [15] and [16], our proposed `tSSNALM` derives from the dual side, rather than the original optimization problem (3). Furthermore, we apply the semi-smooth Newton method to accelerate the augmented Lagrangian method. All resulting subproblems have closed-form solutions, which makes this algorithm efficient for large-scale problems. In the next section, we discuss its convergence properties.

3.4. Convergence Analysis

The first-order optimality conditions of (3) at the local minimizer (β^*, θ^*) are

$$\begin{cases} 0 \in -X^TY\theta^* + \lambda\partial\|\beta^*\|_1 - \langle\omega^*, X^TX\beta^*\rangle, \\ \langle\omega^*, \|X\beta^*\|^2 - 1\rangle = 0, \\ \omega^* \ge 0, \end{cases} \tag{12}$$

and

$$\begin{cases} 0 \in -Y^TX\beta^* + \mu\partial\|\theta^*\|_1 - \langle\nu^*, Y^TY\theta^*\rangle, \\ \langle\nu^*, \|X\beta^*\|^2 - 1\rangle = 0, \\ \nu^* \ge 0. \end{cases} \tag{13}$$

We say that (β^*, θ^*) is a stationary point of (3) if it satisfies (12) and (13). We now establish our convergence result.

Theorem 3.1. *The sequence $\{(\beta^k, \theta^k)\}$ generated by Algorithms 1–3 converges to a stationary point $\{(\beta^*, \theta^*)\}$.*

Proof. Let the objective value be

$$E := -\beta^TX^TY\theta + \lambda\|\beta\|_1 + \mu\|\theta\|_1.$$

On one hand, we have

$$\begin{cases} E_1^k = -(\beta^k)^TX^TY\theta^{k-1} + \lambda\|\beta^k\|_1 + \mu\|\theta^{k-1}\|_1, \\ E_2^k = -(\beta^k)^TX^TY\theta^k + \lambda\|\beta^k\|_1 + \mu\|\theta^k\|_1. \end{cases}$$

Because θ^k is a minimizer of θ-subproblem, we can obtain $E_1^k \ge E_2^k$. On the other hand,

$$\begin{cases} E_2^k = -(\beta^k)^TX^TY\theta^k + \lambda\|\beta^k\|_1 + \mu\|\theta^k\|_1, \\ E_1^{k+1} = -(\beta^{k+1})^TX^TY\theta^k + \lambda\|\beta^{k+1}\|_1 + \mu\|\theta^k\|_1, \end{cases}$$

and similarly $E_2^k \ge E_2^{k+1}$. Combing the above relations, we conclude that

$$E_1^1 \ge E_2^1 \ge E_1^2 \ge \cdots E_1^k \ge E_2^k \ge E_1^{k+1} \ge \cdots.$$

Therefore, the objective sequence $\{(E_1^k, E_2^k)\}$ is decreasing.

Note that updating β and θ in Algorithms 1 is equivalent to alternating projection of β or θ onto two manifolds. Following a similar line of argument as in [29], we can derive its local convergence. □

From the theorem, we can only guarantee local convergence. However, in practice, global solutions are often obtained, as seen from the simulated examples in the next section.

4. Numerical Experiments

In this section, we apply our proposed tSSNALM to the sparse canonical correlation analysis model (2), and compare with the existing methods CoLaR [15] and AMA [16]. All the experiments are performed using MATLAB (R2018b) on a laptop computer with a 1.4 GHz Intel Core i7 CPU and 16 GB of memory.

4.1. Setup

In our numerical tests, we aim to estimate $\beta \in \mathbb{R}^p$ and $\theta \in \mathbb{R}^q$ from the data matrices $X \in \mathbb{R}^{n\times p}$ and $Y \in \mathbb{R}^{n\times q}$ with different choices of triplets (n, p, q), where n is the number of samples, p is the number of features in X, and q is the number of features in Y. We generate the data in the same manner as [15]. Specifically, the data matrices X and Y are generated from

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{pmatrix}\right), \tag{14}$$

where $\Sigma_{xy} = \hat{\rho}\Sigma_x\hat{\beta}\hat{\theta}^T\Sigma_y$ with $\hat{\beta}, \hat{\theta}$ being the true canonical vectors and $\hat{\rho}$ being the true canonical correlation. $\hat{\beta}, \hat{\theta}$ are generated randomly with 5 nonzero entries each at coordinates 1, 6, 11, 16, 21. The nonzeros are obtained from normalizing (with respect to Σ_x and Σ_y) random numbers drawn from a uniform distribution on the finite set $\{-2, -1, 0, 1, 2\}$.

As suggested in [15], structured covariances of x and y are highly investigated in sparse canonical correlation analysis. Thus, we consider three types of covariance matrices in this category $(p = q)$:

- Identity matrices: $\Sigma_x = \Sigma_y = I_p$.
- Toeplitz matrices: $[\Sigma_x]_{ij} = [\Sigma_y]_{ij} = 0.3^{|i-j|}$ for all $i, j \in [p]$.
- Sparse inverse matrices: $[\Sigma_x]_{ij} = [\Sigma_y]_{ij} = \sigma^0_{ij}/\sqrt{\sigma^0_{ii}\sigma^0_{j,j}}$ with $\Sigma^0_x = \Sigma^0_y = (\sigma^0_{ij}) = \Omega^{-1}$ and

$$\Omega_{ij} = \delta_{i=j} + 0.5 \times \delta_{|i-j|=1} + 0.4 \times \delta_{|i-j|=2}, \ \ i, j \in [p].$$

Σ_y is generated in the same fashion.

The matrices X and Y are both divided by $\sqrt{n-1}$ such that X^TY is the estimated covariance matrix. To evaluate the performance of different methods, we apply the discrepancy in [30], i.e.,

$$\text{Error}(\hat{\beta}, \beta) = \min(\|\hat{\beta} - \beta\|^2, \|\hat{\beta} + \beta\|^2) = 2(1 - |\langle\hat{\beta}, \beta\rangle|),$$

where β is the iterate returned by the algorithm, and the second equality follows from the fact that $\|\hat{\beta}\| = \|\beta\| = 1$. In fact, it measures the sine of the angle between $\hat{\beta}$ and β. Moreover, the following procedure for initialization suggested in [16] is adopted. First, we truncate the matrix X^TY by soft-thresholding its small elements to be 0 and denote the resulting matrix S_{xy}. More specifically, we set the entries of X^TY to zero if their magnitude is smaller than the largest magnitude of diagonal elements of X^TY. Secondly, we compute the singular vectors β_0 and θ_0 corresponding to the largest singular value of S_{xy} and then normalize them using

$$\beta_0 := \beta_0/\sqrt{\beta_0^TX^TX\beta_0}, \quad \theta_0 := \theta_0/\sqrt{\theta_0^TY^TY\theta_0}$$

as initialization of β and θ. In addition, the accuracy of the solution depends on the parameters λ and μ. We try values from a set, and choose the ones that give the least Error value.

Finally, we choose ϵ to be 10^{-3}. We terminate the algorithms by checking that the relative change in successive iterates satisfies

$$\max\left\{\frac{\|\beta^{k+1} - \beta^k\|}{\|\beta^k\| + 1}, \frac{\|\theta^{k+1} - \theta^k\|}{\|\theta^k\| + 1}\right\} < 10^{-3}.$$

4.2. Comparison Results

In this section, we compare our proposed tSSNALM with CoLaR [15] and AMA [16]. We implement CoLaR and AMA according to the framework presented in their papers. In order to compare them in the same units, we calculate the estimates of each method and then normalize them by $X\hat{\beta}$, and $Y\hat{\theta}$ respectively. We then normalize estimates such that they all have norm 1. For the sparsity level, we set the number of non-zeros in β and θ to be 1%, with the indices of non-zeros randomly chosen. For the sake of fairness, we run 100 times to illustrate the performance of the algorithms.

We summarize the results in Table 1,2,3 for identity, Toeplitz and sparse inverse covariances, respectively. In each table, the column "Scale" lists different (n, p, q). The column "ρ" refers to the recovered canonical correlation. The columns "Error(β)" and "Error(θ)" report the distance between $\hat{\beta}$ and β, $\hat{\theta}$ and θ, respectively. Last but not least, the columns "CPU" show the running time in seconds. The best method for each test case is highlighted in boldface.

Some observations can be made: (1) In terms of $\hat{\rho}$, Error(β) and Error(θ), tSSNALM performs similarly to CoLaR and AMA. The reason is that they all come from the same SCCA model (2) and make full use of its structure. (2) Compared with

Table 1
Comparison results for identity matrix.

Scale	ρ			Error(β)			Error(θ)			CPU		
(n, p, q)	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM
(100,1000,1000)	0.900	0.900	0.900	1.652e-3	1.652e-3	1.652e-3	2.072e-3	2.071e-3	2.071e-3	0.473	0.366	**0.075**
(100,2000,2000)	0.898	0.898	0.898	2.817e-3	2.815e-3	2.812e-3	3.196e-3	3.196e-3	3.196e-3	0.844	0.612	**0.187**
(100,5000,5000)	0.902	0.902	0.902	1.341e-3	1.341e-3	1.341e-3	2.033e-3	2.033e-3	2.033e-3	1.278	0.998	**0.246**
(100,10000,10000)	0.899	0.899	0.899	1.899e-3	1.899e-3	1.899e-3	2.241e-3	2.241e-3	2.241e-3	2.530	1.664	**0.295**
(200,1000,1000)	0.901	0.901	0.901	2.302e-3	2.302e-3	2.302e-3	1.483e-3	1.483e-3	1.483e-3	1.473	1.243	**0.123**
(200,2000,2000)	0.900	0.900	0.900	1.167e-3	1.167e-3	1.167e-3	2.995e-3	2.995e-3	2.995e-3	1.710	1.535	**0.279**
(200,5000,5000)	0.900	0.900	0.900	3.050e-3	3.050e-3	3.050e-3	2.512e-3	2.512e-3	2.512e-3	2.207	1.981	**0.353**
(200,10000,10000)	0.898	0.898	0.898	1.416e-3	1.416e-3	1.416e-3	1.662e-3	1.662e-3	1.662e-3	3.062	2.784	**0.375**
(500,1000,1000)	0.900	0.900	0.900	1.137e-3	1.137e-3	1.137e-3	3.449e-3	3.449e-3	3.449e-3	2.374	2.366	**0.380**
(500,2000,2000)	0.897	0.897	0.897	1.630e-3	1.630e-3	1.630e-3	1.280e-3	1.280e-3	1.280e-3	2.907	2.843	**0.414**
(500,5000,5000)	0.902	0.902	0.902	2.795e-3	2.795e-3	2.795e-3	2.701e-3	2.701e-3	2.701e-3	3.630	3.286	**0.495**
(500,10000,10000)	0.899	0.899	0.899	1.483e-3	1.483e-3	1.483e-3	2.363e-3	2.363e-3	2.363e-3	4.879	3.697	**0.517**
(1000,1000,1000)	0.898	0.898	0.898	3.611e-3	3.611e-3	3.611e-3	1.304e-3	1.304e-3	1.304e-3	2.981	2.012	**0.478**
(1000,2000,2000)	0.900	0.900	0.900	1.723e-3	1.723e-3	1.723e-3	1.258e-3	1.258e-3	1.258e-3	3.539	3.276	**0.523**
(1000,5000,5000)	0.900	0.900	0.900	2.614e-3	2.614e-3	2.614e-3	2.344e-3	2.344e-3	2.344e-3	4.535	4.270	**0.669**
(1000,10000,10000)	0.899	0.899	0.899	1.700e-3	1.509e-3	1.435e-3	1.205e-3	1.205e-3	1.205e-3	5.705	5.689	**0.675**
(2000,1000,1000)	0.902	0.902	0.902	1.493e-3	1.493e-3	1.493e-3	3.326e-3	3.326e-3	3.326e-3	3.570	2.904	**0.571**
(2000,2000,2000)	0.898	0.898	0.898	4.065e-3	4.065e-3	4.065e-3	1.852e-3	1.852e-3	1.852e-3	4.612	3.009	**0.659**
(2000,5000,5000)	0.900	0.900	0.900	2.872e-3	2.872e-3	2.872e-3	1.766e-3	1.766e-3	1.766e-3	5.375	4.812	**0.762**
(2000,10000,10000)	0.899	0.899	0.899	1.280e-3	1.280e-3	1.280e-3	2.262e-3	2.262e-3	2.262e-3	6.266	5.971	**0.839**

Table 2
Comparison results for Toeplitz matrix.

Scale	ρ			Error(β)			Error(θ)			CPU		
(n, p, q)	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM
(100,1000,1000)	0.902	0.902	0.902	1.717e-3	1.717e-3	1.717e-3	2.441e-3	2.441e-3	2.441e-3	0.537	0.462	**0.062**
(100,2000,2000)	0.900	0.900	0.900	2.514e-3	2.514e-3	2.514e-3	3.345e-3	3.345e-3	3.345e-3	0.842	0.840	**0.138**
(100,5000,5000)	0.890	0.890	0.890	2.034e-3	2.034e-3	2.034e-3	2.360e-3	2.360e-3	2.360e-3	1.391	1.274	**0.306**
(100,10000,10000)	0.896	0.896	0.896	1.808e-3	1.808e-3	1.808e-3	2.992e-3	2.992e-3	2.992e-3	2.011	1.925	**0.788**
(200,1000,1000)	0.901	0.901	0.901	2.541e-3	2.541e-3	2.541e-3	1.578e-3	1.578e-3	1.578e-3	1.246	1.236	**0.290**
(200,2000,2000)	0.898	0.898	0.898	2.580e-3	2.580e-3	2.580e-3	2.906e-3	2.906e-3	2.906e-3	1.592	1.464	**0.325**
(200,5000,5000)	0.899	0.899	0.899	3.154e-3	3.154e-3	3.154e-3	2.435e-3	2.435e-3	2.435e-3	2.171	1.958	**0.436**
(200,10000,10000)	0.901	0.901	0.901	2.980e-3	2.980e-3	2.980e-3	1.712e-3	1.712e-3	1.712e-3	3.420	2.689	**0.491**
(500,1000,1000)	0.901	0.901	0.901	3.567e-3	3.567e-3	3.567e-3	3.490e-3	3.490e-3	3.490e-3	2.488	2.365	**0.374**
(500,2000,2000)	0.900	0.900	0.900	2.342e-3	2.342e-3	2.342e-3	2.293e-3	2.293e-3	2.293e-3	3.053	2.937	**0.563**
(500,5000,5000)	0.903	0.903	0.903	2.795e-3	2.795e-3	2.795e-3	1.508e-3	1.508e-3	1.508e-3	3.354	3.161	**0.597**
(500,10000,10000)	0.897	0.897	0.897	1.329e-3	1.329e-3	1.329e-3	2.789e-3	2.789e-3	2.789e-3	4.523	3.938	**0.669**
(1000,1000,1000)	0.895	0.895	0.895	3.291e-3	3.291e-3	3.291e-3	3.915e-3	3.915e-3	3.915e-3	2.906	2.643	**0.427**
(1000,2000,2000)	0.903	0.903	0.903	2.044e-3	2.044e-3	2.044e-3	1.381e-3	1.381e-3	1.381e-3	3.578	3.119	**0.599**
(1000,5000,5000)	0.901	0.901	0.901	1.314e-3	1.314e-3	1.314e-3	3.452e-3	3.452e-3	3.452e-3	4.924	4.587	**0.785**
(1000,10000,10000)	0.898	0.898	0.898	2.130e-3	2.130e-3	2.130e-3	2.057e-3	2.057e-3	2.057e-3	6.315	5.175	**0.952**
(2000,1000,1000)	0.901	0.901	0.901	1.581e-3	1.581e-3	1.581e-3	1.315e-3	1.315e-3	1.315e-3	3.492	2.999	**0.516**
(2000,2000,2000)	0.899	0.899	0.899	3.570e-3	3.570e-3	3.570e-3	1.942e-3	1.942e-3	1.942e-3	5.287	3.070	**0.697**
(2000,5000,5000)	0.901	0.901	0.901	2.436e-3	2.436e-3	2.436e-3	2.841e-3	2.841e-3	2.841e-3	6.491	5.913	**0.783**
(2000,10000,10000)	0.898	0.898	0.898	1.597e-3	1.597e-3	1.597e-3	3.354e-3	3.354e-3	3.354e-3	8.643	6.125	**0.978**

Table 3
Comparison results for sparse inverse covariances.

Scale	ρ			Error(β)			Error(θ)			CPU		
(n, p, q)	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM	CoLaR	AMA	tSSNALM
(100,1000,1000)	0.899	0.899	0.899	1.831e-3	1.831e-3	1.831e-3	3.216e-3	3.216e-3	3.216e-3	0.548	0.413	**0.055**
(100,2000,2000)	0.900	0.900	0.900	2.119e-3	2.119e-3	2.119e-3	1.103e-3	1.103e-3	1.103e-3	0.933	0.702	**0.138**
(100,5000,5000)	0.902	0.902	0.902	2.472e-3	2.472e-3	2.472e-3	1.385e-3	1.385e-3	1.385e-3	1.358	1.186	**0.179**
(100,10000,10000)	0.898	0.898	0.898	1.231e-3	1.231e-3	1.231e-3	2.363e-3	2.363e-3	2.363e-3	2.625	1.953	**0.286**
(200,1000,1000)	0.901	0.901	0.901	2.337e-3	2.337e-3	2.337e-3	1.412e-3	1.412e-3	1.412e-3	1.456	1.252	**0.144**
(200,2000,2000)	0.903	0.903	0.903	2.548e-3	2.548e-3	2.548e-3	3.360e-3	3.360e-3	3.360e-3	2.012	1.470	**0.282**
(200,5000,5000)	0.900	0.900	0.900	3.252e-3	3.252e-3	3.252e-3	1.481e-3	1.481e-3	1.481e-3	2.853	2.251	**0.347**
(200,10000,10000)	0.897	0.897	0.897	1.417e-3	1.417e-3	1.417e-3	2.279e-3	2.279e-3	2.279e-3	3.599	2.943	**0.465**
(500,1000,1000)	0.902	0.902	0.902	2.625e-3	2.625e-3	2.625e-3	2.533e-3	2.533e-3	2.533e-3	2.014	1.736	**0.498**
(500,2000,2000)	0.899	0.899	0.899	4.179e-3	4.179e-3	4.179e-3	1.128e-3	1.128e-3	1.128e-3	3.460	2.894	**0.565**
(500,5000,5000)	0.900	0.900	0.900	2.286e-3	2.286e-3	2.286e-3	2.560e-3	2.560e-3	2.560e-3	4.223	3.150	**0.681**
(500,10000,10000)	0.898	0.898	0.898	1.109e-3	1.109e-3	1.109e-3	3.082e-3	3.082e-3	3.082e-3	5.958	4.733	**0.803**
(1000,1000,1000)	0.898	0.898	0.898	3.576e-3	3.576e-3	3.576e-3	2.067e-3	2.067e-3	2.067e-3	2.792	2.568	**0.457**
(1000,2000,2000)	0.901	0.901	0.901	2.492e-3	2.492e-3	2.492e-3	1.538e-3	1.538e-3	1.538e-3	3.689	3.752	**0.602**
(1000,5000,5000)	0.900	0.900	0.900	1.676e-3	1.676e-3	1.676e-3	3.356e-3	3.356e-3	3.356e-3	4.725	4.849	**0.711**
(1000,10000,10000)	0.902	0.902	0.902	2.472e-3	2.472e-3	2.472e-3	2.528e-3	2.528e-3	2.528e-3	6.134	5.043	**0.947**
(2000,1000,1000)	0.901	0.901	0.901	1.892e-3	1.892e-3	1.892e-3	1.132e-3	1.132e-3	1.132e-3	3.371	2.636	**0.613**
(2000,2000,2000)	0.899	0.899	0.899	3.128e-3	3.128e-3	3.128e-3	2.274e-3	2.274e-3	2.274e-3	4.681	3.293	**0.794**
(2000,5000,5000)	0.901	0.901	0.901	2.456e-3	2.456e-3	2.456e-3	1.136e-3	1.136e-3	1.136e-3	5.434	4.461	**0.910**
(2000,10000,10000)	0.900	0.900	0.900	2.253e-3	2.253e-3	2.253e-3	1.596e-3	1.596e-3	1.596e-3	7.652	5.047	**1.082**

CoLaR and AMA, the CPU of tSSNALM is always the least. For example, for the identity covariance matrix with $(n, p, q) = (1000, 10000, 10000)$, the CPU time for our proposed model is nearly 8 times faster than that for CoLaR and AMA. This is because the combination of duality, semi-smooth Newton and augmented Lagrangian method makes our algorithm highly efficient.

5. Conclusion and Future Work

We have proposed a fast two-stage semi-smooth Newton augmented Lagrangian method for sparse canonical correlation analysis (SCCA). The key feature, applying a semi-smooth Newton method to the dual problem, can fully exploit its structure. Numerical results have convincingly demonstrated the superior efficiency of our algorithm in solving large-scale SCCA. In future work, we plan to apply tSSNALM to real-world data, such as fault detection, in order to improve the energy efficiency, process monitoring, and sustainability.

Acknowledgments

The authors would like to thank the anonymous referees for their comments and suggestions, which have greatly improved the paper. This work was supported in part by the National Natural Science Foundation of China (61633001, 11671029) and Natural Science Foundation of Shanghai (19ZR1420800).

References

[1] H. Hotelling, Relations between two sets of variates, Biometrika 28 (3-4) (1936) 321–377.
[2] F.V. Waugh, Regressions between sets of variables, Econometrica, Journal of the Econometric Society (1942) 290–310.
[3] M.S. Monmonier, F.E. Finn, Improving the interpretation of geographical canonical correlation models, The Professional Geographer 25 (2) (1973) 140–142.
[4] H. Lindsey, J. Webster, S. Halpern, Canonical correlation as a discriminant tool in a periodontal problem, Biometrical Journal 27 (3) (1985) 257–264.
[5] X.M. Tu, D.S. Burdick, D.W. Millican, L.B. McGown, Canonical correlation technique for rank estimation of excitation-emission matrixes, Analytical Chemistry 61 (19) (1989) 2219–2224.
[6] S.V. Schell, W.A. Gardner, Programmable canonical correlation analysis: A flexible framework for blind adaptive spatial filtering, IEEE Transactions on Signal Processing 43 (12) (1995) 2898–2908.
[7] Z. Chen, S.X. Ding, T. Peng, C. Yang, W. Gui, Fault detection for non-Gaussian processes using generalized canonical correlation analysis and randomized algorithms, IEEE Transactions on Industrial Electronics 65 (2) (2018) 1559–1567.
[8] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correlation analysis, in: International Conference on Machine Learning, 2013, pp. 1247–1255.
[9] V. Uurtio, J.M. Monteiro, J. Kandola, J. Shawe-Taylor, D. Fernandez-Reyes, J. Rousu, A tutorial on canonical correlation methods, ACM Computing Surveys (CSUR) 50 (6) (2018) 95.
[10] N. Martin, H. Maes, Multivariate Analysis, Academic Press, London, 1979.
[11] J. Fan, F. Han, H. Liu, Challenges of big data analysis, National Science Review 1 (2) (2014) 293–314.
[12] R. Tibshirani, M. Wainwright, T. Hastie, Statistical learning with sparsity: the LASSO and generalizations, Chapman and Hall/CRC, 2015.
[13] W. Wang, J. Fan, Asymptotics of empirical eigenstructure for high dimensional spiked covariance, Annals of Statistics 45 (3) (2017) 1342.
[14] D.M. Witten, R. Tibshirani, T. Hastie, A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, Biostatistics 10 (3) (2009) 515–534.
[15] C. Gao, Z. Ma, H.H. Zhou, et al., Sparse CCA: Adaptive estimation and computational barriers, The Annals of Statistics 45 (5) (2017) 2074–2101.
[16] X. Suo, V. Minden, B. Nelson, R. Tibshirani, M. Saunders, Sparse canonical correlation analysis, arXiv preprint arXiv:1705.10865 (2017).
[17] X. Suo, Topics in High-Dimensional Statistical Learning, Stanford University, 2018 Ph.D. thesis.
[18] R.T. Rockafellar, Convex Analysis, Princeton University Press, 2015.
[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends® in Machine Learning 3 (1) (2011) 1–122.
[20] M.R. Hestenes, Multiplier and gradient methods, Journal of Optimization Theory and Applications 4 (5) (1969) 303–320.
[21] M.J.D. Powell, A method for nonlinear constraints in minimization problems, Optimization (1969) 283–298.
[22] J.J. Moreau, Fonctions convexes duales et points proximaux dans un espace Hilbertien, Comptes rendus hebdomadaires des séances de l'Académie des sciences 255 (1961) 2897–2899.
[23] N. Parikh, S. Boyd, et al., Proximal algorithms, Foundations and Trends® in Optimization 1 (3) (2014) 127–239.
[24] D.L. Donoho, De-noising by soft-thresholding, IEEE Transactions on Information Theory 41 (3) (1995) 613–627.
[25] J.-J. Moreau, Proximité et dualité dans un espace Hilbertien, Bulletin de la Société mathématique de France 93 (1965) 273–299.
[26] P.E. Gill, W. Murray, M.A. Saunders, SNOPT: An SQP algorithm for large-scale constrained optimization, SIAM Review 47 (1) (2005) 99–131.
[27] X. Li, D. Sun, K.-C. Toh, A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems, SIAM Journal on Optimization 28 (1) (2018) 433–458.
[28] G.H. Golub, C.F. Van Loan, Matrix Computations, 3, JHU press, 2012.
[29] A.S. Lewis, J. Malick, Alternating projections on manifolds, Mathematics of Operations Research 33 (1) (2008) 216–234.
[30] I.M. Johnstone, A.Y. Lu, On consistency and sparsity for principal components analysis in high dimensions, Journal of the American Statistical Association 104(486) 682–693.