On Sparse Canonical Correlation Analysis

Yongchun Li
University of Tennessee
ycli@utk.edu

Santanu S. Dey
Georgia Tech
santanu.dey@isye.gatech.edu

Weijun Xie
Georgia Tech
wxie@gatech.edu

Abstract

The classical Canonical Correlation Analysis (CCA) identifies the correlations between two sets of multivariate variables based on their covariance, which has been widely applied in diverse fields such as computer vision, natural language processing, and speech analysis. Despite its popularity, CCA can encounter challenges in explaining correlations between two variable sets within high-dimensional data contexts. Thus, this paper studies Sparse Canonical Correlation Analysis (SCCA) that enhances the interpretability of CCA. We first show that SCCA generalizes three well-known sparse optimization problems, sparse PCA, sparse SVD, and sparse regression, which are all classified as NP-hard problems. This result motivates us to develop strong formulations and efficient algorithms. Our main contributions include (i) the introduction of a combinatorial formulation that captures the essence of SCCA and allows the development of approximation algorithms; (ii) the establishment of the complexity results for two low-rank special cases of SCCA; and (iii) the derivation of an equivalent mixed-integer semidefinite programming model that facilitates a specialized branch-and-cut algorithm with analytical cuts. The effectiveness of our proposed formulations and algorithms is validated through numerical experiments.

1 Introduction

The Canonical Correlation Analysis (CCA), proposed by H. Hotelling [23], aims to identify the correlations between two sets of multivariate variables based on their covariance. Since then, CCA has become a powerful statistical technique used for multivariate data analysis, with its applications across diverse fields such as computer vision [24], natural language processing [38], and speech analysis [21]. Despite its popularity, CCA can encounter challenges in explaining correlations between two variable sets within high-dimensional data contexts, such as genomic datasets [36]. In contrast, Sparse Canonical Correlation Analysis (SCCA), which seeks sparse linear combinations of these variable sets, offers substantially enhanced interpretability [41, 42, 44].

Formally, this paper studies the SCCA problem:

$$v^* := \max_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \left\{ \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} : \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \le 1, \boldsymbol{y}^\top \boldsymbol{C}\boldsymbol{y} \le 1, \|\boldsymbol{x}\|_0 \le s_1, \|\boldsymbol{y}\|_0 \le s_2 \right\}, \tag{SCCA}$$

where $s_1 \le n$, $s_2 \le m$ are positive integers and $\begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{C} \end{pmatrix}$ denotes a covariance matrix of $(n+m)$ random variables. Specifically, $\boldsymbol{B}$ and $\boldsymbol{C}$ are the covariance matrices of the n and m random variables, respectively, and $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ is the cross-covariance matrix between n and m random variables. Hence, $\begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{C} \end{pmatrix}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are positive semidefinite matrices of size $(n+m)$, n, and m, respectively. Here, matrices $\boldsymbol{B}, \boldsymbol{C}$ can be singular, i.e., some random variables may be dependent on others. In fact, the covariance matrices $\boldsymbol{B}, \boldsymbol{C}$ are often low-rank, especially within the high-dimension low-sample size data context (see, e.g., the gene expression data in [41]).

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

The SCCA problem generalizes three widely-studied sparsity-constrained optimization problems as special cases, which are sparse PCA [2, 13, 27], sparse SVD [28, 41], and sparse regression [22, 3]. To be specific, when $n = m$, $s_1 = s_2$, $\boldsymbol{B}, \boldsymbol{C}$ are identity matrices, and $\boldsymbol{A}$ is a positive semidefinite matrix, SCCA reduces to the classic sparse PCA problem; when $\boldsymbol{B}, \boldsymbol{C}$ are identity matrices, SCCA becomes the sparse SVD problem; and when $\boldsymbol{A}$ is rank-one, Section 3 shows that SCCA is equivalent to two sparse linear regression subproblems.

1.1 Main contributions

SCCA is generally NP-hard, given that its special cases, sparse PCA, sparse SVD, and sparse regression are all classified as NP-hard problems. We are motivated to develop efficient formulations and algorithms for SCCA through a mixed-integer optimization lens. The main contributions, along with the structure of the remainder of this paper, are the following:

(i) In Section 2, we present an exact semidefinite programming (SDP) reformulation and derive a closed-form optimal value of classic CCA problem. We also develop an equivalent combinatorial formulation of SCCA, which allows the development of approximation algorithms;

(ii) When the covariance matrix $\begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{C} \end{pmatrix}$ is low-rank, Section 3 studies the complexity of two special cases of SCCA. This motivates us to develop a polynomial-time exact algorithm of complexity $\mathcal{O}(n^3 + m^3)$ for solving SCCA to global optimiality when the sparsity levels (i.e., s_1 and s_2) meet or exceed the ranks of $\boldsymbol{B}$ and $\boldsymbol{C}$;

(iii) Section 4 derives an equivalent mixed-integer SDP (MISDP) reformulation for SCCA. When applying the Benders decomposition approach, instead of solving the large-scale SDPs, we design a customized branch-and-cut algorithm with closed-form cuts, which can successfully solve SCCA to optimality; and

(iv) Section 5 numerically test the proposed formulations and algorithms. It is noted that our polynomial-time exact algorithm can solve real-world instances with $n = 19,672$ and $m = 2,149$ variables in seconds, provided that both s_1 and s_2 are at least the ranks of $\boldsymbol{B}$ and $\boldsymbol{C}$.

Our analyses and results can be extended to SCCA with multiple pairs of basis vectors $(\boldsymbol{x}, \boldsymbol{y})$, allowing for a more flexible and comprehensive exploration of correlations among data sets. The detailed formulations of multiple SCCA, along with the computational results, are provided in Appendix F.

1.2 Relevant literature

SCCA. To the best of our knowledge, the work [36] was the first paper that introduced the concept of SCCA to select only small subsets of variables to better explain the relationship between many genetic loci and gene expression phenotypes. A handful subset of features enhances interpretability, a desirable property, especially in complex data analysis, which has been successfully demonstrated in Sparse PCA [25]. To obtain sparse canonical loadings $(\boldsymbol{x}, \boldsymbol{y})$, [39] first applied elastic net penalty to the classical CCA via an iterative regression procedure. In a seminal work on SCCA [41], the authors proposed a rigorous formulation by enforcing the ℓ_1 constraints on variables $(\boldsymbol{x}, \boldsymbol{y})$ and developed a penalized matrix decomposition method to solve the penalized CCA problem. Then, extensive research has focused on various penalty norm functions to obtain sparse canonical loadings (see, e.g., [20, 26, 39, 42, 10]). In particular, [10] penalized multiple canonical loadings by ℓ_1 norm and computed the sparse solution by the linearized Bregman method. It should be noted that under the assumption that the leading canonical loadings are sparse, [7, 17, 18] established theoretical guarantees of iterative approaches for estimating sparse solutions. Another research direction in SCCA introduced penalty functions based on group structural information of input data and developed group SCCA methods [29, 30]. For a comprehensive overview of CCA and SCCA methods, we refer readers to the survey by [44] and the references therein. These approaches, however, do not strictly enforce the exact sparsity requirement but only approximate the sparsity requirement (i.e., the ℓ_0 norm) by a convex function. Another relevant work [40] introduced binary variables to recast SCCA as a mixed-integer nonconvex program under the assumption of positive definite matrices $\boldsymbol{B}, \boldsymbol{C}$, based on which they designed a branch-and-bound algorithm. Different from the literature, our work does not require positive definiteness assumption of matrices $\boldsymbol{B}, \boldsymbol{C}$, and we are able to obtain mixed-integer conic and semidefinite programming reformulations, allowing for better exact and approximation algorithms.

Connections to and differences with sparse PCA and sparse SVD. Analogous to SCCA, both sparse PCA [13, 25] and sparse SVD [28] select small subsets of variables to improve the interpretability of dimensionality reduction methods: PCA and SVD. Considerable investigation has been conducted on solving sparse PCA and sparse SVD from three angles: convex relaxations [12–14], approximation algorithms [6, 9, 28], and exact algorithms [2, 27, 28]. As mentioned before, in sparse PCA and sparse SVD, the covariance matrices $\boldsymbol{B}, \boldsymbol{C}$ are identity. Such a setting dramatically simplifies the subset selection problems of sparse PCA and sparse SVD compared to that of SCCA, as in these problems, it suffices to focus on the selection of a submatrix of the matrix $\boldsymbol{A}$. Specifically, it is shown in [11, 27, 35] that sparse PCA reduces to selecting a principal submatrix of $\boldsymbol{A}$ to maximize the largest eigenvalue(s) and sparse SVD reduces to selecting a possibly non-symmetric submatrix of $\boldsymbol{A}$ to maximize the largest singular value(s) [28]. Quite differently, the combinatorial reformulation (1) of SCCA aims to simultaneously select a sized-$(s_1 \times s_1)$ principal submatrix of $\boldsymbol{B}$, a sized-$(s_2 \times s_2)$ principal submatrix of $\boldsymbol{C}$, and a sized-$(s_1 \times s_2)$ submatrix of $\boldsymbol{A}$. These fundamental differences in the underlying formulations of sparse PCA and sparse SVD preclude the direct application of their existing algorithms to the SCCA.

Notations: The following notation is used throughout the paper. We use bold lower-case letters (e.g., $\boldsymbol{x}$) and bold upper-case letters (e.g., $\boldsymbol{X}$) to denote vectors and matrices, respectively, and we use corresponding non-bold letters (e.g., x_i) to denote their components. We let $\mathcal{S}^n, \mathcal{S}^n_+, \mathcal{S}^n_{++}$ denote the set of all the $n \times n$ symmetric real matrices, the set of all the $n \times n$ symmetric positive semidefinite matrices, and the set of all the $n \times n$ symmetric positive definite matrices, respectively. We let $\boldsymbol{I}$ denote the identity matrix and let $\boldsymbol{0}$ denote the vector or matrix with all-zero entries. We let $\mathbb{R}^n_+$ denote the set of all n-dimensional nonnegative vectors. We let $[n] = \{1, 2, \cdots, n\}$, $[s, n] = \{s, s+1, \cdots, n\}$. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and two subsets $S \subseteq [n]$, $T \subseteq [m]$, we let $\boldsymbol{A}^\dagger$ denote the pseudo inverse of matrix $\boldsymbol{A}$, let $\boldsymbol{A}_{S,T}$ denote a submatrix of $\boldsymbol{A}$ with rows and columns indexed by sets S, T, respectively, and let $(\boldsymbol{A}_{S,T})^\dagger$ denote the pseudo inverse of submatrix $\boldsymbol{A}_{S,T}$. For a set S and an integer k, we define the set $S + k = \{i + k | i \in S\}$. Given a vector $\boldsymbol{a} \in \mathbb{R}^n$ and a subset $S \subseteq [n]$, we let $\boldsymbol{a}_S$ denote a subvector of $\boldsymbol{a}$ in the subset S. We define $[\lambda]_+ = \max\{\lambda, 0\}$. We let $\sigma_{\max}(\cdot)$ denote the largest singular value function and let $\lambda_{\max}(\cdot)$ denote the largest eigenvalue value function.

2 A combinatorial reformulation of SCCA

This section introduces an equivalent combinatorial optimization reformulation of SCCA. This reformulation serves as the foundation for developing two effective approximation algorithms.

2.1 An exact semidefinite programming representation of CCA

To begin with, let us focus on the classic CCA problem, which refers to SCCA without zero-norm constraints, as defined below:

$$\max_{\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in \mathbb{R}^m} \left\{ \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y} : \boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{x} \le 1, \boldsymbol{y}^\top \boldsymbol{C} \boldsymbol{y} \le 1 \right\}. \tag{CCA}$$

This formulation of CCA can be regarded as a quadratically constrained quadratic program concerning the variables $\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \in \mathbb{R}^{n \times m}$. We next define three-block matrices of size $(n+m)$ below that aid in the presentation of our results.

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{A}/2 \\ \boldsymbol{A}^\top/2 & \boldsymbol{0} \end{pmatrix}, \ \tilde{\boldsymbol{B}} = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}, \ \tilde{\boldsymbol{C}} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C} \end{pmatrix}.$$

By introducing a size-$(n+m)$ matrix variable $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix}^\top$ and removing the rank-one constraint on $\boldsymbol{X}$, we can obtain an SDP relaxation of (CCA), as described below

$$\max_{\boldsymbol{X} \in \mathcal{S}^{m+n}_+} \left\{ \operatorname{tr}\left(\tilde{\boldsymbol{A}} \boldsymbol{X}\right) : \operatorname{tr}\left(\tilde{\boldsymbol{B}} \boldsymbol{X}\right) \le 1, \operatorname{tr}\left(\tilde{\boldsymbol{C}} \boldsymbol{X}\right) \le 1 \right\}. \tag{SDP Relaxation}$$

Next, let us present a key lemma regarding properties of block matrices being positive semidefinite, fundamental for reformulating the SCCA.

Lemma 1 ([16]) *For any symmetric matrix* $\begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{C} \end{pmatrix} \in \mathcal{S}^{n+m}$*, the followings are equivalent:*

(i) The block matrix is positive semidefinite;

(ii) $\boldsymbol{B} \in \mathcal{S}_+^n$, $(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^\dagger)\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{C} - \boldsymbol{A}^\top \boldsymbol{B}^\dagger \boldsymbol{A} \in \mathcal{S}_+^m$*; and*

(iii) $\boldsymbol{C} \in \mathcal{S}_+^m$, $(\boldsymbol{I} - \boldsymbol{C}\boldsymbol{C}^\dagger)\boldsymbol{A}^\top = \boldsymbol{0}$, $\boldsymbol{B} - \boldsymbol{A}\boldsymbol{C}^\dagger \boldsymbol{A}^\top \in \mathcal{S}_+^n$.

Inspired by Lemma 1, we hereby establish the equivalence between CCA and its SDP Relaxation. Remarkably, both of these problems achieve the same optimal value, namely $\sigma_{\max}(\sqrt{\boldsymbol{B}^\dagger}\boldsymbol{A}\sqrt{\boldsymbol{C}^\dagger})$.

Proposition 1 *For the CCA problem, we have the following results.*

(i) Both CCA and its SDP Relaxation have an optimal value $\sigma_{\max}(\sqrt{\boldsymbol{B}^\dagger}\boldsymbol{A}\sqrt{\boldsymbol{C}^\dagger})$*;*

(ii) A pair of optimal solutions $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ *to CCA satisfies*

$$\boldsymbol{x}^* = \sqrt{\boldsymbol{B}^\dagger}\boldsymbol{q}, \quad \boldsymbol{y}^* = \sqrt{\boldsymbol{C}^\dagger}\boldsymbol{p},$$

where $\boldsymbol{q} \in \mathbb{R}^n, \boldsymbol{p} \in \mathbb{R}^m$ *denote a pair of leading singular vectors of matrix* $\sqrt{\boldsymbol{B}^\dagger}\boldsymbol{A}\sqrt{\boldsymbol{C}^\dagger}$*; and*

(iii) An optimal solution $\boldsymbol{X}^*$ *to the SDP Relaxation is*

$$\boldsymbol{X}^* = \begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix}^\top .$$

Proof. See Appendix A.1. □

Proposition 1 motivates the following observation on the optimal values of CCA and SCCA.

Observation 1 *The optimal value of CCA is upper bounded by* 1*, so is the optimal value of SCCA.*

It is noteworthy that the results presented in Proposition 1 are established through a distinct methodology. This methodology leverages the positive semidefinite condition of block matrices, as shown in Lemma 1, and incorporates duality theory. This approach differs from most prior research [31, 37, 44], which proved Part (i) of Proposition 1 by relying on the singular value decomposition and assuming that matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ are positive definite (i.e., full rank). To the best of our knowledge, [10] showed parts (i) and (ii) of Proposition 1 for a special low-rank CCA problem, where the authors assumed that the covariance matrices are defined as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}^\top$, $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{U}^\top$, and $\boldsymbol{C} = \boldsymbol{V}\boldsymbol{V}^\top$. Remarkably, Proposition 1 extends this result to a more general scenario where $\boldsymbol{B}$ and $\boldsymbol{C}$ are not constrained to be strictly positive definite and $\boldsymbol{A}$ is not constrained to directly depend on $\boldsymbol{B}, \boldsymbol{C}$, allowing for rank deficiencies and flexible data structure.

2.2 An equivalent formulation of SCCA

In this subsection, we transform SCCA into a combinatorial optimization problem, according to the insights provided by Proposition 1.

Theorem 1 *SCCA is equivalent to the following combinatorial optimization:*

$$v^* = \max_{S_1 \subseteq [m], |S_1| \le s_1, S_2 \subseteq [n], |S_2| \le s_2} \left\{ \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger} \boldsymbol{A}_{S_1,S_2} \sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger} \right) \right\}. \tag{1}$$

Proof. See Appendix A.2. □

The combinatorial formulation (1) presents significant computational difficulties when attempting to solve SCCA. The primary obstacles are two-fold: first, simultaneously selecting submatrices from the matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ requires a sophisticated optimization across multiple dimensions. Second, the selection criterion is particularly complex, as it involves optimizing the largest singular value of the product of the selected submatrix of $\boldsymbol{A}$ and the square root of pseudo-inverse submatrices of $\boldsymbol{B}$ and $\boldsymbol{C}$. These complexities necessitate effective optimization solution procedures to address the high-dimensional and non-convex nature of the problem.

Motivated by Theorem 1, we customize the greedy and local search algorithms for SCCA (1) that has been widely used in the literature to solve special cases of SCCA, such as sparse PCA and sparse SVD in literature (see, e.g., [27, 28]). The detailed implementations can be found in Appendix B.

3 Low-rank SCCA

In practice, it is common that the sample covariance matrix $\begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{C} \end{pmatrix}$ exhibits low-rank characteristics. This phenomenon is especially prominent when dealing with high-dimensional, low-sample size data, e.g., the real gene expression data in [41]. In this section, we study two special cases of low-rank SCCA and their computational complexities. Specifically, we develop a polynomial-time exact algorithm of complexity $\mathcal{O}(n^3 + m^3)$ for solving SCCA to global optimiality when sparsity levels (i.e., s_1 and s_2) exceed or equal the ranks of $\boldsymbol{B}$ and $\boldsymbol{C}$. Besides, we recast SCCA into mixed-integer convex quadratic programming when matrix $\boldsymbol{A}$ is rank-one.

3.1 Special Case I: SCCA with low-rank covariance matrices

In this section, we show that the computational complexity of SCCA is contingent upon the ranks of the covariance matrices $\boldsymbol{B}$ and $\boldsymbol{C}$. To be more precise, when the sparsity level s_1 (or s_2) is equal to or greater than the rank r (or $\hat{r}$) of the covariance matrix $\boldsymbol{B}$ (or $\boldsymbol{C}$), the imposition of a zero-norm constraint over $\boldsymbol{x}$ (or $\boldsymbol{y}$) in SCCA becomes redundant. Consequently, lower ranks in the covariance matrices correspond to better computational complexity in solving SCCA.

Theorem 2 *Suppose $r = \operatorname{rank}(\boldsymbol{B})$ and $\hat{r} = \operatorname{rank}(\boldsymbol{C})$, then SCCA takes a complexity of $\mathcal{O}(n^{r-1}m^{\hat{r}-1} + n^{r-1} + m^{\hat{r}-1})$. The following results hold:*

(i) When $s_1 \geq r$ and $s_2 \geq \hat{r}$, the SCCA problem is equivalent to CCA, i.e.,

$$v^* = \max_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \left\{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} : \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \leq 1, \boldsymbol{y}^\top \boldsymbol{C}\boldsymbol{y} \leq 1\right\}; \tag{2}$$

(ii) When $s_1 \geq r$ and $s_2 < \hat{r}$, the SCCA problem can be reduced to

$$v^* = \max_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \left\{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} : \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \leq 1, \boldsymbol{y}^\top \boldsymbol{C}\boldsymbol{y} \leq 1, \|\boldsymbol{y}\|_0 \leq s_2\right\}; \tag{3}$$

(iii) When $s_1 < r$ and $s_2 \geq \hat{r}$, the SCCA problem can be reduced to

$$v^* = \max_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \left\{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{y} : \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \leq 1, \boldsymbol{y}^\top \boldsymbol{C}\boldsymbol{y} \leq 1, \|\boldsymbol{x}\|_0 \leq s_1\right\}. \tag{4}$$

Proof. See Appendix A.3. □

The results in Theorem 2 build on the covariance structure of the data matrix. Specifically, if B and C are of rank r and $\hat{r}$, respectively, there are only r and $\hat{r}$ linearly independent vectors in the subspaces corresponding to B and C. Thus, the cosine of the principal angle can always be represented by these r and $\hat{r}$ vectors. As a result, the canonical directions of CCA consist of only r and $\hat{r}$ nonzero elements. We further make the following remarks about Theorem 2:

(i) Theorem 2 implies the complexity of solving SCCA, as summarized in the corollary below.

(ii) Inspired by Part (i) of Theorem 2, we also develop a polynomial-time exact algorithm yielding an optimal solution to SCCA (1) when $s_1 \geq r$ and $s_2 \geq \hat{r}$. The detailed implementation can be found in Algorithm 1, which successfully solves some large instances with up to $n = 19,672$ and $m = 2,149$ variables in seconds in our numerical experiments; and

(iii) The proof of Theorem 2 implies that CCA always admits an optimal sparse solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ satisfying $\|\boldsymbol{x}^*\|_0 \leq r$ and $\|\boldsymbol{y}^*\|_0 \leq \hat{r}$. We show in Proposition 1 that the SDP Relaxation of CCA is exact. Therefore, as a side product, we provide the first-known sufficient condition about (i.e., $s_1 \geq r$ and $s_2 \geq \hat{r}$) when the convex SDP Relaxation matches SCCA.

Corollary 1 *Suppose $r = \mathrm{rank}(\boldsymbol{B})$ and $\hat{r} = \mathrm{rank}(\boldsymbol{C})$. There exists an algorithm that can find an optimal solution to SCCA in $\mathcal{O}(n^{r-1}m^{\hat{r}-1})$ time complexity.*

Proposition 2 *Suppose $r = \mathrm{rank}(\boldsymbol{B})$ and $\hat{r} = \mathrm{rank}(\boldsymbol{C})$. Then Algorithm 1 returns an optimal solution to SCCA (1) in $\mathcal{O}(n^3 + m^3)$ time complexity when $s_1 \geq r$ and $s_2 \geq \hat{r}$.*

Proof. Following the proof of Theorem 2, we can show that the output solution of Algorithm 1 is optimal to SCCA (1). In addition, the Step 2 of Algorithm 1 needs computing the eigendecomposition of matrix $\boldsymbol{B} \in \mathcal{S}_+^n$, which takes a time of $\mathcal{O}(n^3)$. Given a matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times(n-r)}$, it also takes a time of $\mathcal{O}(n^3)$ to find its $(n-r)$ linearly independent rows at Step 3 through the QR decomposition [19]. Hence, Algorithm 1 takes a complexity of $\mathcal{O}(n^3 + m^3)$. □

Algorithm 1 An exact algorithm for SCCA (1) when $s_1 \geq r$ and $s_2 \geq \hat{r}$

1: **Input:** Matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{B} \in \mathcal{S}_+^m$, $\boldsymbol{C} \in \mathcal{S}_+^m$ and integers $s_1 \in [r, n]$, $s_2 \in [\hat{r}, m]$
2: Compute the eigenvectors $\boldsymbol{Q} \in \mathbb{R}^{n\times(n-r)}$ of $\boldsymbol{B}$ that correspond to its $(n-r)$ zero eigenvalues,
3: Find $(n-r)$ linearly independent rows in $\boldsymbol{Q}$, and collect their indices into a subset $T_1^* \subseteq [n]$
4: Perform the same procedure on matrix $\boldsymbol{C}$ to obtain the subset $T_2^* \subseteq [m]$
5: Define the subsets $S_1^* = [n] \setminus T_1^*$ and $S_2^* = [m] \setminus T_2^*$, and compute

$$v^* = \sigma_{\max}\left(\sqrt{\left(\boldsymbol{B}_{S_1^*,S_1^*}\right)^\dagger}\boldsymbol{A}_{S_1^*,S_2^*}\sqrt{\left(\boldsymbol{C}_{S_2^*,S_2^*}\right)^\dagger}\right)$$

6: **Output:** An optimal solution (S_1^*, S_2^*) and optimal value v^*

3.2 Special Case II: SCCA with a rank-one cross-covariance matrix

In this subsection, we study the other interesting low-rank special case of SCCA where the cross-covariance matrix $\boldsymbol{A}$ is rank-one. For this special case, we prove its NP-hardness with reduction to the sparse regression problem. We further demonstrate that rank-one SCCA can be simplified to solving two Mixed-Integer Convex Quadratic Programs (MICQPs), which can be more scalable than directly solving SCCA. Our numerical findings confirm this improved scalability.

We observe that SCCA can be separable over variables $\boldsymbol{x}$ and $\boldsymbol{y}$ for the rank-one $\boldsymbol{A}$. In fact, suppose that $\boldsymbol{A} = \boldsymbol{a}\boldsymbol{b}^\top$, then SCCA is equivalent to

$$v^* = \max_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \left\{\boldsymbol{x}^\top\boldsymbol{a}\boldsymbol{b}^\top\boldsymbol{y} : \boldsymbol{x}^\top\boldsymbol{B}\boldsymbol{x} \leq 1, \boldsymbol{y}^\top\boldsymbol{C}\boldsymbol{y} \leq 1, \|\boldsymbol{x}\|_0 \leq s_1, \|\boldsymbol{y}\|_0 \leq s_2\right\} \tag{5}$$

which can be equivalently the product of the optimal values of the following two subproblems:

$$\begin{aligned} v_x &= \max_{\boldsymbol{x}\in\mathbb{R}^n}\{\boldsymbol{a}^\top\boldsymbol{x} : \boldsymbol{x}^\top\boldsymbol{B}\boldsymbol{x} \leq 1, \|\boldsymbol{x}\|_0 \leq s_1\}, \\ v_y &= \max_{\boldsymbol{y}\in\mathbb{R}^m}\{\boldsymbol{b}^\top\boldsymbol{y} : \boldsymbol{y}^\top\boldsymbol{C}\boldsymbol{y} \leq 1, \|\boldsymbol{y}\|_0 \leq s_2\}. \end{aligned} \tag{6}$$

That is, the identity $v^* = v_x v_y$ holds. Next, we show that each subproblem in (6) can be reduced to the classic sparse regression problem [1, 33] and is thus NP-hard as shown below.

Theorem 3 *When matrix $\boldsymbol{A} = \boldsymbol{a}^\top\boldsymbol{b}$ is rank-one, each maximization problem in (6) is NP-hard.*

Proof. See Appendix A.4. □

Theorem 3 links the maximization problem (6) and the well-known sparse regression problem, implying that even solving the rank-one SCCA problem (5) is NP-hard. However, it also motivates us to adapt existing mixed-integer optimization techniques from sparse regression (see, e.g., [1, 4, 43]) to tackle each subproblem in (6). By introducing binary variables to model the zero-norm constraint, we derive equivalent MICQP formulations for subproblems (6) in Appendix C. There are two types of formulations depending on whether matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ are positive definite, which build on the Big-M and perspective techniques, respectively.

4 Reformulating SCCA as a mixed-integer semidefinite program (MISDP)

While the combinatorial formulation (1) is elegant in its structure, it poses significant challenges when attempting to solve it to optimality using branch-and-bound based methods. To fill this gap, in this section, we derive an equivalent MISDP formulation for SCCA, amenable for developing exact methods.

First, it is noted that an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ to SCCA is always bounded that satisfies $\|\boldsymbol{x}^*\|_2^2 \leq M_1$ and $\|\boldsymbol{y}^*\|_2^2 \leq M_2$, where we specify the construction of the coefficients M_1 and M_2 in Appendix C.1. Such bounds are essential to the derivation of the MISDP. It is convenient to introduce the following notation about $\{M_{ii}\}_{i\in[n+m]}$:

$$M_{ii} = M_1, \forall i \in [n], \ \ M_{ii} = M_2, \forall i \in [n+1, n+m].$$

Theorem 4 *The SCCA is equivalent to the following MISDP:*

$$v^{=} \max_{\boldsymbol{X}\in\mathcal{S}_+^{n+m}, \boldsymbol{z}\in\mathcal{Z}} \{\mathrm{tr}(\tilde{\boldsymbol{A}}\boldsymbol{X}) : \mathrm{tr}(\tilde{\boldsymbol{B}}\boldsymbol{X}) \leq 1, \mathrm{tr}(\tilde{\boldsymbol{C}}\boldsymbol{X}) \leq 1, X_{ii} \leq M_{ii} z_i, \forall i \in [n+m]\}. \tag{7}$$

where the feasible set is defined as $\mathcal{Z} = \{\boldsymbol{z} \in \{0,1\}^{n+m} : \sum_{i\in[n]} z_i \leq s_1, \sum_{i\in[n+1,n+m]} z_i \leq s_2\}$.

Proof. See Appendix A.5. □

Note that the proposed MISDP formulation (7) is of size $(n+m)\times(n+m)$ since our matrix variable $\boldsymbol{X}$ replaces $\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{pmatrix}^\top$ in SCCA.

We have formulated SCCA as a mixed-integer convex optimization problem in Theorem 4. Unfortunately, no commercial solvers can efficiently solve MISDP problems. We derive an equivalent mixed-integer linear program of SCCA with exponentially many linear constraints and an efficient separation oracle based on the approach introduced by [15], which allows us to develop a tailored branch-and-cut algorithm. First, by separating the binary variables $\boldsymbol{z}$, we rewrite the MISDP (7) as

$$v^* = \max_{\boldsymbol{z}\in\mathcal{Z}, v} \{v : v \leq f(\boldsymbol{z})\}, \tag{8}$$

where the function $f(\boldsymbol{z})$ is defined as

$$f(\boldsymbol{z}) = \max_{\boldsymbol{X}\in\mathcal{S}_+^{n+m}} \left\{\mathrm{tr}(\tilde{\boldsymbol{A}}\boldsymbol{X}) : \mathrm{tr}(\tilde{\boldsymbol{B}}\boldsymbol{X}) \leq 1, \mathrm{tr}(\tilde{\boldsymbol{C}}\boldsymbol{X}) \leq 1, X_{ii} \leq M_{ii} z_i, \forall i \in [n+m]\right\}. \tag{9}$$

For any feasible solution $\hat{\boldsymbol{z}} \in \mathcal{Z}$ of the problem (8), by leveraging the concavity of function $f(\cdot)$, the linear inequality

$$v \leq f(\hat{\boldsymbol{z}}) + \partial f(\hat{\boldsymbol{z}})^\top(\boldsymbol{z} - \hat{\boldsymbol{z}})$$

cuts off the solution $\hat{\boldsymbol{z}}$ unless it happened to be optimal in (8), which paves the way for a delayed cut-generation procedure within a branch-and-bound framework. As the linear inequality of the above type needs to be added dynamically given different solutions $\hat{\boldsymbol{z}}$ at each iteration, it calls for an efficient evaluation of function $f(\hat{\boldsymbol{z}})$ and its subgradient. To speed up the computation, we derive the closed-form expression for both of them. The detailed derivations can be found in Appendix D.

Strategies to improve computational speed in practice: First, we provide a variable-fixing method that can identify some binary variables of the MISDP (7) being one at optimality. Removing these pre-selected variables from the feasible set reduces the problem size of SCCA. Second, we enhance the branch-and-cut algorithm with a high-quality warm start solution obtained from the local search algorithm. Third, by relaxing the binary variables in the MISDP (7) to be continuous or computing CCA, we can obtain an upper bound of SCCA, and the gap between this bound and the local search output gives an initial gap at the root node. Finally, at each iteration, the branching node is selected based on its potential to decrease the current upper bound instead of random branching.

5 Numerical results

This section tests the numerical performance of our formulations and algorithms on synthetic and real data. All the experiments are conducted in Python 3.6 with calls to Gurobi 9.5.2 and MOSEK 10.0.29 on a PC with 10-core CPU, 16-core GPU, and 16GB of memory. The codes and data used in our experiments are available at `https://github.com/yongchunli-13/SCCA.git`.

5.1 Experimental setup

Synthetic data generation: Before we present the empirical results, we first describe the properties of the synthetic data which shall be used throughout this section. By following [32], given parameters (n, m, s_1, s_2), we first synthetically generate positive definite matrices $\boldsymbol{B}^* \in \mathcal{S}_{++}^n$ and $\boldsymbol{C}^* \in \mathcal{S}_{++}^n$ by $\boldsymbol{B}^* = \hat{\boldsymbol{B}}\hat{\boldsymbol{B}}^\top + \boldsymbol{I}$ and $\boldsymbol{C}^* = \hat{\boldsymbol{C}}\hat{\boldsymbol{C}}^\top + \boldsymbol{I}$, respectively, where $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{C}}$ consist of elements generated from a normal distribution $\mathcal{N}(0, 1)$. Then, we let $\boldsymbol{A}^* \in \mathbb{R}^{n \times m} = \lambda \boldsymbol{B}^* \boldsymbol{u}\boldsymbol{v}^\top \boldsymbol{C}^*$, where we generate λ uniformly from $(0, 1)$, and vectors $\boldsymbol{u}, \boldsymbol{v}$ are generated from a normal distribution $\mathcal{N}(0, 1)$ that satisfy $\|\boldsymbol{u}\|_0 = s_1$, $\|\boldsymbol{v}\|_0 = s_2$, $\boldsymbol{u}^\top \boldsymbol{B}^* \boldsymbol{u} = 1$ and $\boldsymbol{v}^\top \boldsymbol{C}^* \boldsymbol{v} = 1$. Next, we sample $N = 5,000$ data samples from a normal distribution $\mathcal{N}\left(\boldsymbol{0}, \begin{pmatrix} \boldsymbol{B}^* & \boldsymbol{A}^* \\ (\boldsymbol{A}^*)^\top & \boldsymbol{C}^* \end{pmatrix}\right)$ and compute their sample covariance matrix to obtain the testing data $\begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{C} \end{pmatrix}$.

Real data: To obtain a comprehensive understanding of the overall performance of our algorithms, we further conduct experiments on six UCI datasets [5] with sizes ranging from 34 to 385 variables. The dataset is split into the first n variables and the remaining m variables to construct the sample covariance matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$. Besides, we examine the performance of the proposed algorithms on the real breast cancer dataset [8] that contain $n = 19,672$ and $m = 2,149$ variables. The information on each dataset is summarized in Appendix E.

Throughout, the computational time is in seconds, the time limit is one hour, and the dashed line "-" denotes the unsolved case within the time limit. Note that we let **LB** denote the lower bound obtained from the approximation algorithm, and we let **UB** denote the upper bound obtained from convex relaxations of SCCA. Besides, we define **gap(%)**$= 100 \times (\text{UB} - v^*)/v^*$ to be the optimality gap, and we replace v^* with the best lower bound when v^* is not available. We define **MIPGap(%)** to be the gap of exact algorithms at termination. Notably, the complexity analysis of the SCCA problems in Section 3 indicates that its solution process depends on the ranks of the data matrices. Therefore, we present the numerical results under both full-rank and low-rank cases for a comprehensive evaluation.

5.2 Illustration of the impact sparsity levels on SCCA

In this subsection, we apply the local search algorithm to evaluating the performance of SCCA against different sparsity levels s_1 and s_2. Specifically, for a given dataset, we compute the ratio of correlations between SCCA and CCA for various s_1, s_2 parameters. We test the real UCI data and synthetic data, and the results are displayed in Figure 1 and Figure 2. This visualization provides insights for the maximum sparsity SCCA can achieve while maintaining the correlation of full data. For the real UCI data, SCCA almost recovers the correlation of CCA when $s_1 \approx n/2$ and $s_2 \approx m/2$.

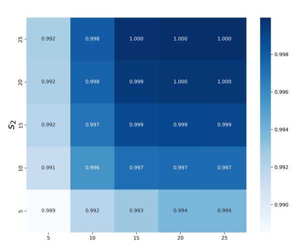

Figure 1: On UCI data with $n, m = 28, 29$

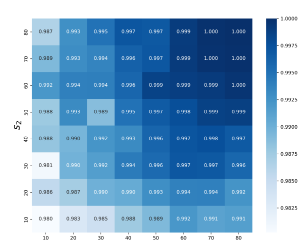

Figure 2: On synthetic data with $n, m = 80, 80$

5.3 Solving SCCA with full-rank matrices

The numerical results on synthetic and real data are presented in Table 1 and Table 2, respectively, which include multiple instances with various parameters (n, m, s_1, s_2). First, we observe that the greedy and local search algorithms are scalable, and their outputs match the optimal values for most solved testing cases. That is, they achieve zero optimality gaps on these cases. In the "Convex relaxation" column, we compute an upper bound by solving either the continuous relaxation of MISDP (7) or CCA, and we use CCA for $n \geq 40$ and $m \geq 40$ cases for efficiency. It is seen that the upper bound maintains an optimality gap at most 2.78%. Then, we apply the branch-and-cut algorithm to solve SCCA to optimality, which can handle the case up to a size of $n = m = 120$ in

Table 1. The unsolved case in Table 1 may be because the initial gap is weak at the root node, implying that the branch-and-cut algorithm explores a considerable amount of nodes before termination (see, e.g., $n = m = 80$ and $s_1 = s_2 = 10$). Hence, the branch-and-cut algorithm may struggle with large-scale instances with weak initial gaps.

Table 1: Evaluation of algorithms on synthetic data

				Greedy		Local search		Convex relaxation			Branch-and-cut		
n	m	s_1	s_2	LB	time	LB	time	UB	gap(%)	time	v^*	MIPGap(%)	time
20	20	5	5	0.244	0.01	0.244	0.02	0.256	1.23	1	0.244	0.00	9
20	20	10	10	0.275	0.02	0.275	0.04	0.278	1.23	1	0.275	0.00	4
40	40	5	5	0.695	0.03	0.695	0.05	0.701	0.83	1	0.695	0.00	1
40	40	10	10	0.705	0.06	0.705	0.12	0.708	0.45	1	0.705	0.00	7
60	60	5	5	0.885	0.04	0.885	0.09	0.887	0.28	1	0.885	0.00	1
60	60	10	10	0.884	0.09	0.884	0.19	0.887	0.28	1	0.884	0.00	8
80	80	5	5	0.633	0.06	0.633	0.12	0.650	2.78	1	0.633	0.00	1705
80	80	10	10	0.631	0.13	0.631	0.26	0.644	2.02	1	0.643	1.85	-
100	100	5	5	0.942	0.09	0.942	0.16	0.944	0.23	1	0.942	0.00	4
100	100	10	10	0.940	0.17	0.940	0.34	0.942	0.23	1	0.940	0.00	15
120	120	5	5	0.845	0.11	0.845	0.27	0.853	0.97	1	0.845	0.00	924
120	120	10	10	0.848	0.21	0.848	0.43	0.856	0.85	1	0.855	0.80	-

Table 2: Evaluation of algorithms on six UCI datasets

				Greedy		Local search		Convex relaxation			Branch-and-cut		
n	m	s_1	s_2	LB	time	LB	time	$\hat{v}$	gap(%)	time	v^*	MIPGap(%)	time
17	17	5	5	0.970	0.01	0.971	0.04	0.984	1.35	1	0.980	0.00	1
17	17	10	10	0.981	0.02	0.983	0.14	0.984	0.09	1	0.983	0.00	1
28	29	5	5	0.761	0.02	0.761	0.05	0.769	1.11	1	0.761	0.00	6
28	29	10	10	0.766	0.04	0.766	0.07	0.769	0.45	1	0.766	0.00	22
32	32	5	5	0.991	0.02	0.991	0.04	0.993	0.17	1	0.991	0.00	1
32	32	10	10	0.992	0.04	0.992	0.12	0.993	0.05	1	0.992	0.00	1
38	39	5	5	1	0.02	0.02	0.06	0.16	0.00	0.01	1	0.00	2
38	39	10	10	1	0.04	0.05	0.07	0.47	0.00	0.01	1	0.00	2
64	64	5	5	0.998	0.05	0.998	0.39	0.999	0.07	1	0.998	0.00	1
64	64	10	10	0.999	0.09	0.999	0.59	0.999	0.02	1	0.999	0.00	2
192	193	5	5	1	0.21	1	0.46	1	0.00	1	1	0.00	36
192	193	10	10	1	0.36	1	0.68	1	0.00	1	1	0.00	37

5.4 Solving SCCA with low-rank matrices

Despite the high dimensions of the breast cancer dataset [8], the resulting sample covariance matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ have a rank of 89, i.e., $r = \hat{r} = 89$. When $s_1 \geq r$ and $s_2 \geq \hat{r}$, we apply Algorithm 1 to solving SCCA that returns optimal solutions and values in one second, as shown in Table 3. If $s_1 < r$ and $s_2 < \hat{r}$, Algorithm 1 cannot applied and the proposed branch-and-cut algorithm is hard to scale to this dataset. Hence, we only consider using approximation algorithms to solve SCCA for small s_1 and s_2 cases in Table 4. To be specific, we randomly sample n and m variables from the breast cancer data to construct testing cases. As displayed in Table 4, the greedy and local search algorithms run fast, and the local search algorithm slightly outperforms the greedy output.

Table 3: Solving SCCA by Algorithm 1 on breast cancer data when $s_1 \geq r$ and $s_2 \geq \hat{r}$

				Algorithm 1		
n	m	s_1	s_2	v^*	MIPGap(%)	time
19,672	2,149	100	100	1	0.00	1
		150	150	1	0.00	1
		200	200	1	0.00	1
		250	250	1	0.00	1

Table 4: Approximation algorithms on breast cancer data when $s_1 \leq r$ and $s_2 \leq \hat{r}$

				Greedy		Local search	
n	m	s_1	s_2	LB	time	LB	time
100	100	10	10	0.983	0.17	0.985	0.63
500	500	10	10	0.991	1	0.993	6
800	800	10	10	0.993	5	0.993	37
1000	1000	20	20	1	12	1	146

The SCCA (5) problem with a rank-one matrix $\boldsymbol{A}$ can be more tractable, as we can equivalently decompose it into two MICQPs (see Appendix C). By approximating $\boldsymbol{A}$ with a rank-one matrix including leading singular value and vectors, Table 5 presents the numerical results for solving rank-one SCCA (5). The continuous relaxation of the MICQPs also provides an upper bound and is denoted by the **Perspective relaxation** in Table 5. We see that the perspective relaxation is computationally efficient and yields small optimality gaps. Besides, we can directly solve two MICQPs below via Gurobi to find the optimal value of rank-one SCCA (5), i.e., $v^* = v_x v_y$, where the performance can be found in the last column of Table 5. We can address the rank-one SCCA (5) problem up to size 200×200 within one hour, improving the problem-solving capacity compared to the size-120×120 full-rank SCCA (5) in Table 1. However, it should be pointed out that the SCCA may not be mixed-integer convex quadratic representable in general.

Table 5: Solving SCCA on synthetic data with a rank-one matrix $\boldsymbol{A}$

				Greedy		Local Search		Perspective relaxation			SCCA (5)		
n	m	s_1	s_2	LB	time	LB	time	UB	gap(%)	time	v^*	MIPGap(%)	time
50	50	5	5	0.753	0.04	0.753	0.07	0.757	0.57	1	0.753	0.00	1
50	50	10	10	0.753	0.07	0.753	0.15	0.757	0.46	1	0.753	0.00	9
100	100	5	5	0.975	0.09	0.975	0.16	0.977	0.21	1	0.975	0.00	2
100	100	10	10	0.966	0.17	0.966	0.33	0.969	0.35	1	0.966	0.00	25
150	150	5	5	0.850	0.15	0.850	0.26	0.859	1.08	1	0.850	0.00	9
150	150	10	10	0.857	0.27	0.857	0.53	0.867	1.13	1	0.857	0.00	167
200	200	5	5	0.810	0.23	0.810	0.37	0.828	2.19	2	0.810	0.00	55
200	200	10	10	0.816	0.39	0.816	0.74	0.833	2.11	2	0.816	0.00	1692

5.5 Experimental comparison of SCCA algorithms

This section compares the proposed local search algorithm with the SCCA methods of [10, 37, 41] in correlation (Corr) value, the zero norm of $\boldsymbol{x}$ (denoted by S.$\boldsymbol{x}$), the zero norm of $\boldsymbol{y}$ (denoted by S.$\boldsymbol{y}$), and running time. The computational results on synthetic, UCI, and breast cancer data are presented in Table 6, where we highlight the best correlation and sparsity results in bold. Unlike the local search algorithm, these existing methods do not strictly enforce the exact sparsity requirement, i.e., the zero-norm constraints on variables $\boldsymbol{x}, \boldsymbol{y}$. Consequently, the local search algorithm achieves the best sparsity for nearly all testing cases. More importantly, the local search algorithm yields a larger correlation value than these existing methods in 16 out of 22 testing cases. Finally, the running time of the local search algorithm dominates that of [10, 37].

Table 6: Comparison of SCCA algorithms in Correlation, sparsity, and time

				Local search				[41]				[37]				[10]			
n	m	s_1	s_2	Corr	S.$\boldsymbol{x}$	S.$\boldsymbol{y}$	time	Corr	S.$\boldsymbol{x}$	S.$\boldsymbol{y}$	time	Corr	S.$\boldsymbol{x}$	S.$\boldsymbol{y}$	time	Corr	S.$\boldsymbol{x}$	S.$\boldsymbol{y}$	time
20	20	5	5	0.244	**5**	**5**	0.04	0.200	9	11	0.01	0.239	13	14	2	**0.256**	18	16	15
20	20	10	10	0.275	**10**	**10**	0.06	0.212	7	8	0.01	0.259	19	14	2	**0.278**	18	18	12
40	40	5	5	0.695	**5**	**5**	0.08	0.594	23	30	0.03	0.659	26	19	2	**0.696**	17	18	13
40	40	10	10	**0.705**	**10**	**10**	0.17	0.597	17	15	0.01	0.660	19	22	2	0.704	16	20	15
60	60	5	5	**0.885**	**5**	**5**	0.13	0.794	37	45	0.01	0.865	27	25	3	0.880	10	8	23
60	60	10	10	**0.884**	**10**	**10**	0.27	0.777	33	36	0.01	0.847	31	31	3	0.879	12	15	23
80	80	5	5	0.633	**5**	**5**	0.17	0.534	55	43	0.03	0.606	36	32	3	**0.634**	33	23	34
80	80	10	10	0.631	**10**	**10**	0.37	0.528	53	38	0.02	0.603	29	41	3	**0.632**	36	41	32
100	100	5	5	**0.942**	**5**	**5**	0.22	0.813	58	63	0.02	0.902	46	33	4	0.941	5	5	38
100	100	10	10	**0.940**	**10**	**10**	0.49	0.768	54	55	0.02	0.884	47	37	4	0.938	10	10	37
120	120	5	5	**0.845**	**5**	**5**	0.42	0.681	83	74	0.02	0.804	37	43	4	0.833	10	8	47
120	120	10	10	**0.848**	**10**	**10**	0.59	0.720	71	76	0.02	0.821	39	36	4	0.840	13	12	47
17	17	5	5	**0.971**	**5**	**5**	0.04	0.794	6	9	0.01	0.742	10	8	2	0.970	14	16	15
28	29	5	5	**0.761**	5	**5**	0.05	0.704	23	29	0.01	0.667	**3**	7	2	0.744	24	19	15
32	32	5	5	**0.991**	**5**	**5**	0.04	0.730	31	23	0.01	0.904	15	10	2	0.906	18	15	43
38	39	5	5	**1**	**5**	**5**	0.15	0.994	8	9	0.43	0.997	38	23	3	1	24	32	29
64	64	5	5	**0.998**	**5**	**5**	0.38	0.897	64	64	0.01	0.990	7	6	2	0.997	33	31	19
192	193	5	5	**1**	**5**	**5**	0.45	0.103	82	50	0.21	0.856	13	12	10	1	11	6	45
100	100	10	10	0.985	**10**	**10**	0.63	0.936	18	41	0.01	0.686	42	43	4	0.952	75	76	36
500	500	10	10	**0.993**	**10**	**10**	6	0.977	78	200	0.78	0.871	173	371	31	0.935	74	44	50
800	800	10	10	**0.993**	**10**	**10**	37	0.974	133	316	1.94	0.879	280	526	50	0.990	84	63	57
1000	1000	10	10	**0.995**	**10**	**10**	11	0.980	166	401	7.30	0.848	401	558	58	0.985	80	59	60

Acknowledgments and Disclosure of Funding

Santanu S. Dey would like to gratefully acknowledge the support from ONR grant # N00014-22-1-2632. Yongchun Li (when was a graduate student) and Weijun Xie were supported in part by the National Science Foundation grant # 2246417 and ONR grant # N00014-24-1-2066. The authors also would like to gratefully acknowledge undergraduate researcher Quill O Healey for his help with the initial version of branch-and-bound codes.

References

[1] Alper Atamturk and Andres Gomez. Rank-one convexification for sparse regression. *arXiv preprint arXiv:1901.10334*, 2019.

[2] Dimitris Bertsimas and Ryan Cory-Wright. Solving large-scale sparse PCA to certifiable (near) optimality. *The Journal of Machine Learning Research*, 23(1):566–600, 2022.

[3] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813 – 852, 2016.

[4] Dimitris Bertsimas, Jean Pauphilet, and Bart Van Parys. Rejoinder: Sparse regression: Scalable algorithms and empirical performance. *Statistical Science*, 35(4):623–624, 2020.

[5] Catherine L Blake. UCI repository of machine learning databases. *http://www. ics. uci. edu/˜ mlearn/MLRepository. html*, 1998.

[6] Siu On Chan, Dimitris Papailliopoulos, and Aviad Rubinstein. On the approximability of sparse PCA. In *Conference on Learning Theory*, pages 623–646. PMLR, 2016.

[7] Mengjie Chen, Chao Gao, Zhao Ren, and Harrison H Zhou. Sparse cca via precision adjusted iterative thresholding. *arXiv preprint arXiv:1311.6186*, 2013.

[8] Koei Chin, Sandy DeVries, Jane Fridlyand, Paul T Spellman, Ritu Roydasgupta, Wen-Lin Kuo, Anna Lapuk, Richard M Neve, Zuwei Qian, Tom Ryder, et al. Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer cell*, 10(6):529–541, 2006.

[9] Agniva Chowdhury, Petros Drineas, David P Woodruff, and Samson Zhou. Approximation algorithms for sparse principal component analysis. *arXiv preprint arXiv:2006.12748*, 2020.

[10] Delin Chu, Li-Zhi Liao, Michael K Ng, and Xiaowei Zhang. Sparse canonical correlation analysis: New formulation and algorithm. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):3050–3065, 2013.

[11] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(7), 2008.

[12] Alexandre d'Aspremont, Laurent Ghaoui, Michael Jordan, and Gert Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *Advances in neural information processing systems*, 17, 2004.

[13] Santanu S Dey, Rahul Mazumder, and Guanyi Wang. Using ℓ_1-relaxation and integer programming to obtain dual bounds for sparse PCA. *Operations Research*, 70(3):1914–1932, 2022.

[14] Santanu S Dey, Marco Molinaro, and Guanyi Wang. Solving sparse principal component analysis with global support. *Mathematical Programming*, 199(1-2):421–459, 2023.

[15] Marco A Duran and Ignacio E Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36:307–339, 1986.

[16] Jean Gallier et al. The schur complement and symmetric positive semidefinite (and definite) matrices (2019). *URL https://www. cis. upenn. edu/jean/schur-comp. pdf*, 2020.

[17] Chao Gao, Zongming Ma, Zhao Ren, and Harrison H Zhou. Minimax estimation in sparse canonical correlation analysis. 2015.

[18] Chao Gao, Zongming Ma, and Harrison H Zhou. Sparse cca: Adaptive estimation and computational barriers. 2017.

[19] Colin R Goodall. 13 computation using the qr decomposition. 1993.

[20] David R Hardoon and John Shawe-Taylor. Sparse canonical correlation analysis. *Machine Learning*, 83:331–353, 2011.

[21] Hynek Hermansky and Nelson Morgan. Rasta processing of speech. *IEEE transactions on speech and audio processing*, 2(4):578–589, 1994.

[22] Ronald R Hocking and RN Leslie. Selection of the best subset in regression analysis. *Technometrics*, 9(4):531–540, 1967.

[23] Harold Hotelling. The most predictable criterion. *Journal of educational Psychology*, 26(2):139, 1935.

[24] Hua Huang, Huiting He, Xin Fan, and Junping Zhang. Super-resolution of human face image using canonical correlation analysis. *Pattern Recognition*, 43(7):2532–2543, 2010.

[25] John NR Jeffers. Two case studies in the application of principal component analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 16(3):225–236, 1967.

[26] Kim-Anh Lê Cao, Pascal GP Martin, Christèle Robert-Granié, and Philippe Besse. Sparse canonical methods for biological data integration: application to a cross-platform study. *BMC bioinformatics*, 10:1–17, 2009.

[27] Yongchun Li and Weijun Xie. Exact and approximation algorithms for sparse PCA. *arXiv preprint arXiv:2008.12438*, 2020.

[28] Yongchun Li and Weijun Xie. Beyond symmetry: Best submatrix selection for the sparse truncated svd. *arXiv preprint arXiv:2105.03179*, 2021.

[29] Dongdong Lin, Vince D Calhoun, and Yu-Ping Wang. Correspondence between fmri and snp data by group sparse canonical correlation analysis. *Medical image analysis*, 18(6):891–902, 2014.

[30] Dongdong Lin, Jigang Zhang, Jingyao Li, Vince D Calhoun, Hong-Wen Deng, and Yu-Ping Wang. Group sparse canonical correlation analysis for genomic data integration. *BMC bioinformatics*, 14(1):1–16, 2013.

[31] Yichao Lu and Dean P Foster. Large scale canonical correlation analysis with iterative least squares. *Advances in Neural Information Processing Systems*, 27, 2014.

[32] Qing Mai and Xin Zhang. An iterative penalized least squares approach to sparse canonical correlation analysis. *Biometrics*, 75(3):734–744, 2019.

[33] Alan Miller. *Subset selection in regression*. CRC Press, 2002.

[34] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[35] Dimitris Papailiopoulos, Alexandros Dimakis, and Stavros Korokythakis. Sparse PCA through low-rank approximations. In *International Conference on Machine Learning*, pages 747–755. PMLR, 2013.

[36] Elena Parkhomenko, David Tritchler, and Joseph Beyene. Genome-wide sparse canonical correlation of gene expression with genotypes. In *BMC proceedings*, volume 1, pages 1–5. BioMed Central, 2007.

[37] Elena Parkhomenko, David Tritchler, and Joseph Beyene. Sparse canonical correlation analysis with application to genomic data integration. *Statistical applications in genetics and molecular biology*, 8(1), 2009.

[38] Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. *Advances in neural information processing systems*, 15, 2002.

[39] Sandra Waaijenborg, Philip C Verselewel de Witt Hamer, and Aeilko H Zwinderman. Quantifying the association between gene expressions and dna-markers by penalized canonical correlation analysis. *Statistical applications in genetics and molecular biology*, 7(1), 2008.

[40] Akihisa Watanabe, Ryuta Tamura, Yuichi Takano, and Ryuhei Miyashiro. Branch-and-bound algorithm for optimal sparse canonical correlation analysis. *Expert Systems with Applications*, 217:119530, 2023.

[41] Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[42] Daniela M Witten and Robert J Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical applications in genetics and molecular biology*, 8(1), 2009.

[43] Weijun Xie and Xinwei Deng. Scalable algorithms for the sparse ridge regression. *SIAM Journal on Optimization*, 30(4):3359–3386, 2020.

[44] Xinghao Yang, Weifeng Liu, Wei Liu, and Dacheng Tao. A survey on canonical correlation analysis. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2349–2368, 2019.

Appendices

A Proofs of technical results

A.1 Proof of Proposition 1

Proof. The proof includes three parts.

Part (i). To prove the equivalence between CCA and its SDP Relaxation, let us introduce the Lagrangian multiplies $\theta_1 \geq 0, \theta_2 \geq 0$ corresponding to two constraints in SDP Relaxation, which leads to the following Lagrangian dual problem

$$\min_{\theta_1 \geq 0, \theta_2 \geq 0} \left\{ \theta_1 + \theta_2 : \theta_1 \tilde{\boldsymbol{B}} + \theta_2 \tilde{\boldsymbol{C}} \succeq \tilde{\boldsymbol{A}} \right\} = \min_{\theta_1 \geq 0, \theta_2 \geq 0} \left\{ \theta_1 + \theta_2 : \begin{pmatrix} \theta_1 \boldsymbol{B} & \frac{\boldsymbol{A}}{-2} \\ \frac{\boldsymbol{A}^\top}{-2} & \theta_2 \boldsymbol{C} \end{pmatrix} \succeq 0 \right\} \tag{10}$$

where the equation results from the definition of block matrices $\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}$, and $\tilde{\boldsymbol{C}}$. Given the nonzero matrices $\boldsymbol{A} \neq \boldsymbol{0}, \boldsymbol{B} \neq \boldsymbol{0}, \boldsymbol{C} \neq \boldsymbol{0}$ and positive semidefinite matrices $\boldsymbol{B} \succeq 0, \boldsymbol{C} \succeq 0$, following Lemma 1, we must have $\theta_2 \boldsymbol{C} - \boldsymbol{A}^\top (\theta_1 \boldsymbol{B})^\dagger \boldsymbol{A}/4 \succeq 0$ and $\theta_1 \boldsymbol{B} - \boldsymbol{A}(\theta_2 \boldsymbol{C})^\dagger \boldsymbol{A}^\top/4 \succeq 0$, implying that either $\theta_1 = 0$ or $\theta_2 = 0$ is infeasible to the minimization problem above. That is, $\theta_1 > 0$ and $\theta_2 > 0$ must hold.

According to Lemma 1, the block matrix $\begin{pmatrix} \boldsymbol{B} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{C} \end{pmatrix}$ is positive semidefinite, implying that $(\boldsymbol{I} - \boldsymbol{C}\boldsymbol{C}^\dagger)\boldsymbol{A}^\top = \boldsymbol{0}, (\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^\dagger)\boldsymbol{A} = \boldsymbol{0}$. Then, it is easy to show

$$\left(\boldsymbol{I} - \theta_2 \boldsymbol{C}(\theta_2 \boldsymbol{C})^\dagger\right) \frac{\boldsymbol{A}^\top}{2} = \boldsymbol{0}, \forall \theta_2 > 0.$$

Given $\theta_1, \theta_2 > 0$ and using Lemma 1, the result above allows us to further simplify the right-hand side minimization problem in (10) to

$$\begin{aligned}
&\min_{\theta_1 \geq 0, \theta_2 \geq 0} \left\{ \theta_1 + \theta_2 : 4\theta_1\theta_2 \boldsymbol{B} \succeq \boldsymbol{A}\boldsymbol{C}^\dagger \boldsymbol{A}^\top \right\} \\
&= \min_{\theta_1 \geq 0, \theta_2 \geq 0} \left\{ \theta_1 + \theta_2 : 4\theta_1\theta_2 \geq \sigma_{\max}^2 \left(\sqrt{\boldsymbol{B}^\dagger} \boldsymbol{A} \sqrt{\boldsymbol{C}^\dagger} \right) \right\} = \sigma_{\max} \left(\sqrt{\boldsymbol{B}^\dagger} \boldsymbol{A} \sqrt{\boldsymbol{C}^\dagger} \right),
\end{aligned}$$

where the first equation is because

$$\begin{aligned}
&4\theta_1\theta_2 \boldsymbol{B} \succeq \boldsymbol{A}\boldsymbol{C}^\dagger \boldsymbol{A}^\top \iff 4\theta_1\theta_2 \boldsymbol{I} \succeq \sqrt{\boldsymbol{\Lambda}^{-1}} \boldsymbol{Q}^\top \boldsymbol{A}\boldsymbol{C}^\dagger \boldsymbol{A}^\top \boldsymbol{Q} \sqrt{\boldsymbol{\Lambda}^{-1}} \\
&\iff 4\theta_1\theta_2 \geq \lambda_{\max} \left(\sqrt{\boldsymbol{\Lambda}^{-1}} \boldsymbol{Q}^\top \boldsymbol{A}\boldsymbol{C}^\dagger \boldsymbol{A}^\top \boldsymbol{Q} \sqrt{\boldsymbol{\Lambda}^{-1}} \right) \\
&\iff 4\theta_1\theta_2 \geq \lambda_{\max} \left(\sqrt{\boldsymbol{C}^\dagger} \boldsymbol{A}^\top \boldsymbol{B}^\dagger \boldsymbol{A} \sqrt{\boldsymbol{C}^\dagger} \right) \iff 4\theta_1\theta_2 \geq \sigma_{\max}^2 \left(\sqrt{\boldsymbol{B}^\dagger} \boldsymbol{A} \sqrt{\boldsymbol{C}^\dagger} \right),
\end{aligned}$$

where we let $\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ denote the eigendecomposition of matrix $\boldsymbol{B}$ with $\boldsymbol{\Lambda}$ containing all the positive eigenvalues.

As a result, the dual problem of SDP Relaxation admits an optimal value of $\sigma_{\max}\left(\sqrt{\boldsymbol{B}^\dagger} \boldsymbol{A} \sqrt{\boldsymbol{C}^\dagger}\right)$, which gives an upper bound of the CCA and its SDP Relaxation. Next, we construct their optimal solutions, which exactly attain this upper bound. Thus, this upper bound is achievable and equals their optimal values.

Part (ii). For the CCA, let us consider a part of optimal solutions $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ below

$$\boldsymbol{x}^* = \sqrt{\boldsymbol{B}^\dagger} \boldsymbol{q}, \;\; \boldsymbol{y}^* = \sqrt{\boldsymbol{C}^\dagger} \boldsymbol{p},$$

with $\boldsymbol{q} \in \mathbb{R}^n, \boldsymbol{p} \in \mathbb{R}^m$ denoting a pair of leading singular vectors of matrix $\sqrt{\boldsymbol{B}^\dagger} \boldsymbol{A} \sqrt{\boldsymbol{C}^\dagger}$.

First, $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is feasible to the CCA as

$$(\boldsymbol{x}^*)^\top \boldsymbol{B} \boldsymbol{x}^* = \boldsymbol{q}^\top \sqrt{\boldsymbol{B}^\dagger} \boldsymbol{B} \sqrt{\boldsymbol{B}^\dagger} \boldsymbol{q} \leq \boldsymbol{q}^\top \boldsymbol{q} = 1, \;\; (\boldsymbol{y}^*)^\top \boldsymbol{C} \boldsymbol{y}^* = \boldsymbol{p}^\top \sqrt{\boldsymbol{C}^\dagger} \boldsymbol{C} \sqrt{\boldsymbol{C}^\dagger} \boldsymbol{p} \leq \boldsymbol{p}^\top \boldsymbol{p} = 1,$$

where the inequalities stem from the facts that $\boldsymbol{I} \succeq \sqrt{\boldsymbol{B}^\dagger} \boldsymbol{B} \sqrt{\boldsymbol{B}^\dagger}$ and $\boldsymbol{I} \succeq \sqrt{\boldsymbol{C}^\dagger} \boldsymbol{C} \sqrt{\boldsymbol{C}^\dagger}$.

On the other hand, according to the definitions of $\boldsymbol{q}, \boldsymbol{p}$, we can show that $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is optimal to the CCA, i.e.,

$$(\boldsymbol{x}^*)^\top \boldsymbol{A}\boldsymbol{y}^* = \boldsymbol{q}^\top \sqrt{\boldsymbol{B}^\dagger}\boldsymbol{A}\sqrt{\boldsymbol{C}^\dagger}\boldsymbol{p} = \sigma_{\max}\left(\sqrt{\boldsymbol{B}^\dagger}\boldsymbol{A}\sqrt{\boldsymbol{C}^\dagger}\right).$$

Part (iii). In a similar vein, we can show that $\boldsymbol{X}^* = \begin{pmatrix}\boldsymbol{x}^*\\ \boldsymbol{y}^*\end{pmatrix}\begin{pmatrix}\boldsymbol{x}^*\\ \boldsymbol{y}^*\end{pmatrix}^\top$ is optimal to SDP Relaxation with the optimal value $\sigma_{\max}\left(\sqrt{\boldsymbol{B}^\dagger}\boldsymbol{A}\sqrt{\boldsymbol{C}^\dagger}\right)$. □

A.2 Proof of Theorem 1

Proof. By introducing the subsets (S_1, S_2) to denote the supports of variables $(\boldsymbol{x}, \boldsymbol{y})$ in SCCA, then we can remove the zero-norm constraints on $(\boldsymbol{x}, \boldsymbol{y})$ and reformulate SCCA as

$$v^* = \max_{\substack{S_1\subseteq[m],|S_1|\le s_1,\\ S_2\subseteq[n],|S_2|\le s_2}} \max_{\substack{\boldsymbol{x}\in\mathbb{R}^{|S_1|},\\ \boldsymbol{y}\in\mathbb{R}^{|S_2|}}} \left\{\boldsymbol{x}^\top \boldsymbol{A}_{S_1,S_2}\boldsymbol{y} : \boldsymbol{x}^\top \boldsymbol{B}_{S_1,S_1}\boldsymbol{x} \le 1, \boldsymbol{y}^\top \boldsymbol{C}_{S_2,S_2}\boldsymbol{y} \le 1\right\}. \tag{11}$$

Following from the Part (i) in Proposition 1, we can show that for any subsets $S_1 \subseteq [n], S_2 \subseteq [m]$, the following identity holds.

$$\max_{\boldsymbol{x}\in\mathbb{R}^{|S_1|},\boldsymbol{y}\in\mathbb{R}^{|S_2|}} \left\{\boldsymbol{x}^\top \boldsymbol{A}_{S_1,S_2}\boldsymbol{y} : \boldsymbol{x}^\top \boldsymbol{B}_{S_1,S_1}\boldsymbol{x} \le 1, \boldsymbol{y}^\top \boldsymbol{C}_{S_2,S_2}\boldsymbol{y} \le 1\right\}$$
$$= \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger}\boldsymbol{A}_{S_1,S_2}\sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger}\right).$$

Plugging the result above into the inner maximization problem in (11), we complete the proof. □

A.3 Proof of Theorem 2

Proof. The proof is split into three parts.

Part (i). It suffices to prove that CCA admits an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ satisfying $\|\boldsymbol{x}^*\|_0 \le r$ and $\|\boldsymbol{y}^*\|_0 \le \hat{r}$. Then, $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is also feasible and optimal to SCCA, which implies the equivalence between SCCA and CCA.

First, according to Part (ii) in Proposition 1, we can obtain a closed-form optimal solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ for the CCA. By adjusting $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, we will construct a new optimal sparse solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ satisfying $\|\boldsymbol{x}^*\|_0 \le r$ and $\|\boldsymbol{y}^*\|_0 \le \hat{r}$.

For matrix $\boldsymbol{B} \in \mathcal{S}_+^n$, we let $\{\boldsymbol{q}_i\}_{i\in[n-r]} \in \mathbb{R}^n$ denote the eigenvectors corresponding to $(n-r)$ zero eigenvalues of $\boldsymbol{B}$. Thus, $\{\boldsymbol{q}_i\}_{i\in[n-r]}$ are orthonormal. There exists a size-$(n-r)$ subset $S \subseteq [n]$ such that the subvectors $\{(\boldsymbol{q}_i)_S\}_{i\in[n-r]}$ are linearly independent, where $(\boldsymbol{q}_i)_S$ denotes the subvector of $\boldsymbol{q}_i$ indexed by S for each $i \in [n-r]$. As a result, there exist a vector $(\gamma_1, \cdots, \gamma_{n-r})^\top$ such that

$$\hat{\boldsymbol{x}}_S = \sum_{i\in[n]} \gamma_i (\boldsymbol{q}_i)_S. \tag{12}$$

Let us now construct solution $\boldsymbol{x}^*$

$$\boldsymbol{x}^* = \hat{\boldsymbol{x}} - \sum_{i\in[n-r]} \gamma_i \boldsymbol{q}_i,$$

where $x_i^* = 0$ for all $i \in S$ based on the equation (12) and $|S| = n-r$, implying $\|\boldsymbol{x}^*\|_0 \le r$. In addition, we show that the new solution $\boldsymbol{x}^*$ is still optimal to CCA. First, $\boldsymbol{x}^*$ is feasible since

$$(\boldsymbol{x}^*)^\top \boldsymbol{B}(\boldsymbol{x}^*) = \hat{\boldsymbol{x}}^\top \boldsymbol{B}\hat{\boldsymbol{x}} \le 1,$$

where the equation is due to $\boldsymbol{B}\boldsymbol{q}_i = \boldsymbol{0}$ for all $i \in [n-r]$.

Given the positive semidefinite block matrix $\begin{pmatrix}\boldsymbol{B} & \boldsymbol{A}\\ \boldsymbol{A}^\top & \boldsymbol{C}\end{pmatrix}$, using Part (ii) of Lemma 1, the identity $(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^\dagger)\boldsymbol{A} = \boldsymbol{0}$ is equivalent to $\sum_{i\in[n-r]} \boldsymbol{q}_i\boldsymbol{q}_i^\top \boldsymbol{A} = \boldsymbol{0}$. Then, for each $i \in [n-r]$, multiplying

$\boldsymbol{q}_i^\top$ on both sides of this equation leads to

$$\boldsymbol{q}_i^\top \Big(\sum_{j \in [n-r]} \boldsymbol{q}_j \boldsymbol{q}_j^\top \boldsymbol{A} \Big) \boldsymbol{A} = \boldsymbol{q}_i^\top \boldsymbol{0} \Longrightarrow \boldsymbol{q}_i^\top \boldsymbol{A} = \boldsymbol{0},$$

where the result follows from $\boldsymbol{q}_i^\top \boldsymbol{q}_j = 0$ for any $i \neq j$. Then, we can show the optimality of the new solution $\boldsymbol{x}^*$:

$$(\boldsymbol{x}^*)^\top \boldsymbol{A}\hat{\boldsymbol{y}} = \hat{\boldsymbol{x}}^\top \boldsymbol{A}\hat{\boldsymbol{y}} + \sum_{i \in [n-r]} \beta_i \boldsymbol{q}_i^\top \boldsymbol{A}\hat{\boldsymbol{y}} = \hat{\boldsymbol{x}}^\top \boldsymbol{A}\hat{\boldsymbol{y}}.$$

Similarly, we can also construct an optimal sparse solution $\boldsymbol{y}^*$ by leveraging $\hat{\boldsymbol{y}}$ and eigenvectors of zero eigenvalues of $\boldsymbol{C}$ such that $\|\boldsymbol{y}^*\|_0 \leq s_2$.

Therefore, there exists an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ to the CCA whose zero norms are bounded from above by $r, \hat{r}$, respectively. Adding the constraints $\|\boldsymbol{x}\|_0 \leq r, \|\boldsymbol{y}\|_0 \leq \hat{r}$ to the CCA does not affect the optimality, which gives an equivalent formulation (2) of CCA.

Part (ii). Suppose that $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ denotes an optimal solution to problem (3). When $s_1 \geq r$, following the proof of Part (I), $\tilde{\boldsymbol{x}}$, we can construct another optimal solution $\boldsymbol{x}^*$ whose zero norm is bounded by r and $(\boldsymbol{x}^*, \tilde{\boldsymbol{y}})$ is feasible and optimal to SCCA.

Part (iii). Similarly, we can reduce SCCA to problem (4). We thus complete the proof. □

A.4 Proof of Theorem 3

Proof. Let us first consider the maximization problem over $\boldsymbol{x}$ in (6), i.e.,

$$v_x = \max_{\boldsymbol{x} \in \mathbb{R}^n} \{\boldsymbol{a}^\top \boldsymbol{x} : \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \leq 1, \|\boldsymbol{x}\|_0 \leq s_1\}. \tag{13}$$

Then, we derive a combinatorial optimization reformulation of problem (13) based on the result below.

Claim 1 *For any subset $S \subseteq [n]$, $\max_{\boldsymbol{x} \in \mathbb{R}^{|S|}} \{\boldsymbol{a}_S^\top \boldsymbol{x} : \boldsymbol{x}^\top \boldsymbol{B}_{S,S} \boldsymbol{x} \leq 1\} = \sqrt{\boldsymbol{a}_S^\top (\boldsymbol{B}_{S,S})^\dagger \boldsymbol{a}_S}$.*

Proof. Given $\boldsymbol{A} = \boldsymbol{a}\boldsymbol{b}^\top$, since the matrix $\begin{pmatrix} \boldsymbol{B} & \boldsymbol{a}\boldsymbol{b}^\top \\ \boldsymbol{b}^\top \boldsymbol{a} & \boldsymbol{C} \end{pmatrix}$ is positive semidefinite, using Lemma 1, the identity $(\boldsymbol{I} - \boldsymbol{B}_{S,S}\boldsymbol{B}_{S,S}^\dagger)\boldsymbol{a}_S \boldsymbol{b}^\top = \boldsymbol{0}$ must hold for any subset S. As a result, we have $\boldsymbol{a}_S - \boldsymbol{B}_{S,S}\boldsymbol{B}_{S,S}^\dagger \boldsymbol{a}_S = \boldsymbol{0}$ as vector $\boldsymbol{b}$ is nonzero.

Next, the Lagrangian dual of the problem $\max_{\boldsymbol{x} \in \mathbb{R}^{|S|}} \{\boldsymbol{a}_S^\top \boldsymbol{x} : \boldsymbol{x}^\top \boldsymbol{B}_{S,S} \boldsymbol{x} \leq 1\}$ can be written as

$$\begin{aligned}
\max_{\boldsymbol{x} \in \mathbb{R}^{|S|}} \{\boldsymbol{a}_S^\top \boldsymbol{x} : \boldsymbol{x}^\top \boldsymbol{B}_{S,S} \boldsymbol{x} \leq 1\} &= \min_{\mu \geq 0} \max_{\boldsymbol{x} \in \mathbb{R}^{|S|}} \boldsymbol{a}_S^\top \boldsymbol{x} + \mu - \mu \boldsymbol{x}^\top \boldsymbol{B}_{S,S} \boldsymbol{x} \\
&= \min_{\mu \geq 0} \mu + \frac{\boldsymbol{a}_S^\top \boldsymbol{B}_{S,S}^\dagger \boldsymbol{a}_S}{4\mu} = \sqrt{\boldsymbol{a}_S^\top (\boldsymbol{B}_{S,S})^\dagger \boldsymbol{a}_S},
\end{aligned}$$

where the second equation builds on the identity $\boldsymbol{a}_S - \boldsymbol{B}_{S,S}\boldsymbol{B}_{S,S}^\dagger \boldsymbol{a}_S = \boldsymbol{0}$ and optimal solution $\boldsymbol{x}^* = \frac{\boldsymbol{B}_{S,S}^\dagger \boldsymbol{a}_S}{\sqrt{\boldsymbol{a}_S^\top (\boldsymbol{B}_{S,S})^\dagger \boldsymbol{a}_S}}$. ◇

Suppose that an optimal solution to problem (13) admits the support S^*. According to Claim 1, we have

$$v_x = \max_{S \subseteq [n], |S| \leq s} \sqrt{\boldsymbol{a}_S^\top (\boldsymbol{B}_{S,S})^\dagger \boldsymbol{a}_S} = \sqrt{\boldsymbol{a}_{S^*}^\top (\boldsymbol{B}_{S^*,S^*})^\dagger \boldsymbol{a}_{S^*}}.$$

On the other hand, the Lagrangian dual of problem (13) can be written as

$$\begin{aligned}
v_x &\leq \min_{\lambda \in \mathbb{R}_+} \max_{\boldsymbol{x} \in \mathbb{R}^n} \{\boldsymbol{a}^\top \boldsymbol{x} + \lambda - \lambda \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} : \|\boldsymbol{x}\|_0 \leq s_1\} \\
&= \min_{\lambda \in \mathbb{R}_+} \max_{S \subseteq [n], |S| \leq s} \lambda + \frac{\boldsymbol{a}_S^\top (\boldsymbol{B}_{S,S})^\dagger \boldsymbol{a}_S}{4\lambda} \\
&\leq \max_{S \subseteq [n], |S| \leq s} \lambda^* + \frac{\boldsymbol{a}_S^\top (\boldsymbol{B}_{S,S})^\dagger \boldsymbol{a}_S}{4\lambda^*} = \sqrt{\boldsymbol{a}_{S^*}^\top (\boldsymbol{B}_{S^*,S^*})^\dagger \boldsymbol{a}_{S^*}} \leq v_x,
\end{aligned}$$

where the first equation is due to Claim 1, the second inequality is by plugging the feasible solution $\lambda^* = \frac{\sqrt{\boldsymbol{a}_{S^*}^\top (\boldsymbol{B}_{S^*,S^*})^\dagger \boldsymbol{a}_{S^*}}}{2}$ into minimization problem, and the last equation is from the optimality of subset S^*. Since both left-hand and right-hand sides above equal v_x, the strong duality of problem (13) holds, and all the inequalities above must attain the equalities. That is, problem (13) is equivalent to

$$v_x = \min_{\lambda \in \mathbb{R}_+} \max_{\boldsymbol{x} \in \mathbb{R}^n} \{\boldsymbol{a}^\top \boldsymbol{x} + \lambda - \lambda \boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{x} : \|\boldsymbol{x}\|_0 \le s_1\}.$$

Since the outer minimization is a one-dimensional convex program that can be solved efficiently, as a result, for any given $\lambda > 0$, the inner maximization is equivalent to solving

$$\max_{\boldsymbol{x} \in \mathbb{R}^n} \{\boldsymbol{a}^\top \boldsymbol{x} - \lambda \boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{x} : \|\boldsymbol{x}\|_0 \le s_1\}. \tag{14}$$

Next, let us consider the NP-hard sparse regression problem (see, e.g., [34]), which admits

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \left\{\|\boldsymbol{v} - \boldsymbol{U}\boldsymbol{x}\|_2^2 : \|\boldsymbol{x}\|_0 \le s\right\} \Longleftrightarrow \max_{\boldsymbol{x} \in \mathbb{R}^n} \left\{2\boldsymbol{v}^\top \boldsymbol{U}\boldsymbol{x} - \boldsymbol{x}^\top \boldsymbol{U}^\top \boldsymbol{U} \boldsymbol{\beta} : \|\boldsymbol{x}\|_0 \le s\right\}, \tag{15}$$

where data matrix $\boldsymbol{U}$ consists of observations of n variables and vector $\boldsymbol{v}$ denotes the corresponding response variables.

Suppose that in the problem (14), let us define $\lambda \boldsymbol{B} = \boldsymbol{U}^\top \boldsymbol{U}$ and $\boldsymbol{a} = 2\boldsymbol{U}^\top \boldsymbol{v}$. Then using the singular value decomposition of matrix $\boldsymbol{U}$, we see that the following equation still holds.

$$\boldsymbol{a}_S - \boldsymbol{B}_{S,S} \boldsymbol{B}_{S,S}^\dagger \boldsymbol{a}_S = \boldsymbol{0}, \forall S \subseteq [n].$$

Thus, for any given $\lambda > 0$, the maximization problem (14) is equivalent to the sparse regression problem (15). This shows that problem (13) is NP-hard.

Similarly, the maximization problem over $\boldsymbol{y}$ in (6) can also be reduced to the sparse regression problem. $\square$

A.5 Proof of Theorem 4

Proof. For the SCCA (11), according to Proposition 1, the inner maximization problem admits an exact semidefinite programming formulation. Using the variables $\boldsymbol{z} \in \mathcal{Z}$ to describe the set constraints in SCCA (11), we can reformulate it as

$$\begin{aligned} v^* = \max_{\boldsymbol{z} \in \mathcal{Z}} \max_{\boldsymbol{X} \in \mathcal{S}_+^{n+m}} \Big\{ & \operatorname{tr}(\tilde{\boldsymbol{A}}\boldsymbol{X}) : \operatorname{tr}(\tilde{\boldsymbol{B}}\boldsymbol{X}) \le 1, \operatorname{tr}(\tilde{\boldsymbol{C}}\boldsymbol{X}) \le 1, \\ & X_{ii}(1 - z_i) = 0, \forall i \in [m+n]\Big\}. \end{aligned} \tag{16}$$

Proposition 3 shows that there is an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ to SCCA that satisfies $\|\boldsymbol{x}^*\|_2^2 \le M_1$ and $\|\boldsymbol{y}^*\|_2^2 \le M_2$. Based on this, we can construct an optimal solution $(\boldsymbol{z}^*, \boldsymbol{X}^*)$ for SCCA (16) by letting

$$\boldsymbol{X}^* = \begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix}^\top, z_i = \begin{cases} 1 & \text{if } x_i^* \ne 0 \\ 0 & \text{if } x_i^* = 0 \end{cases}, \forall i \in [n], z_{i+n} = \begin{cases} 1 & \text{if } y_i^* \ne 0 \\ 0 & \text{if } y_i^* = 0 \end{cases}, \forall i \in [m],$$

where the optimal solution $\boldsymbol{X}^*$ satisfies the following inequalities

$$X_{ii}^* = (x_i^*)^2 \le M_1 z_i, \forall i \in [n], \ \ X_{(i+n)(i+n)}^* = (y_i^*)^2 \le M_2 z_{i+n}, \forall i \in [m].$$

This allows us to recast the SCCA (16) into an MISDP formulation (7). $\square$

B Implementations of greedy and local search algorithms

This section presents the detailed implementations of greedy and local search algorithms based on the combinatorial formulation (1) of SCCA.

C Mixed-integer convex quadratic programming reformulations

This section shows that each subproblem in (6) can be equivalently formulated by a MICQP. Therefore, SCCA (5) is mixed-integer convex quadratic representable when matrix $\boldsymbol{A}$ is rank-one.

Algorithm 2 Greedy algorithm for SCCA (1)

Input: Matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{B} \in \mathcal{S}_+^m$, $\boldsymbol{C} \in \mathcal{S}_+^m$ and integers $s_1 \in [n]$, $s_2 \in [m]$
Compute $(i^*, j^*) \in \operatorname{argmax}_{i\in[m], j\in[n]} \sqrt{(B_{ii})^\dagger} A_{ij} \sqrt{(C_{jj})^\dagger}$
Define subsets $\hat{S}_1 = \{i^*\}$ and $\hat{S}_2 = \{j^*\}$
for $\ell = 2, \cdots, \max\{s_1, s_2\}$ **do**
  if $\ell \le \min\{s_1, s_2\}$ **then**
    $i^* \in \operatorname{argmax}_{i\in[n]\setminus\hat{S}_1} \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{\hat{S}_1\cup\{i\}, \hat{S}_1\cup\{i\}})^\dagger} \boldsymbol{A}_{\hat{S}_1\cup\{i\}, \hat{S}_2} \sqrt{(\boldsymbol{C}_{\hat{S}_2, \hat{S}_2})^\dagger}\right)$
    Update $\hat{S}_1 = \hat{S}_1 \cup \{i^*\}$
    $j^* \in \operatorname{argmax}_{j\in[m]\setminus\hat{S}_2} \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{\hat{S}_1, \hat{S}_1})^\dagger} \boldsymbol{A}_{\hat{S}_1, \hat{S}_2\cup\{j\}} \sqrt{(\boldsymbol{C}_{\hat{S}_2\cup\{j\}, \hat{S}_2\cup\{j\}})^\dagger}\right)$
  else if $s_1 \le s_2$ **then**
    $j^* \in \operatorname{argmax}_{j\in[m]\setminus\hat{S}_2} \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{\hat{S}_1, \hat{S}_1})^\dagger} \boldsymbol{A}_{\hat{S}_1, \hat{S}_2\cup\{j\}} \sqrt{(\boldsymbol{C}_{\hat{S}_2\cup\{j\}, \hat{S}_2\cup\{j\}})^\dagger}\right)$
    Update $\hat{S}_2 = \hat{S}_2 \cup \{j^*\}$
  else
    $i^* \in \operatorname{argmax}_{i\in[n]\setminus\hat{S}_1} \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{\hat{S}_1\cup\{i\}, \hat{S}_1\cup\{i\}})^\dagger} \boldsymbol{A}_{\hat{S}_1\cup\{i\}, \hat{S}_2} \sqrt{(\boldsymbol{C}_{\hat{S}_2, \hat{S}_2})^\dagger}\right)$
    Update $\hat{S}_1 = \hat{S}_1 \cup \{i^*\}$
  end if
end for
Output: $\hat{S}_1, \hat{S}_2$

Algorithm 3 Local search algorithm for SSVD (1)

Input: Matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{B} \in \mathcal{S}_+^m$, $\boldsymbol{C} \in \mathcal{S}_+^m$ and integers $s_1 \in [n]$, $s_2 \in [m]$
Initialize $(\hat{S}_1, \hat{S}_2)$ as the output of greedy Algorithm 2
do
for each pair $(i_1, j_1) \in \hat{S}_1 \times ([n] \setminus \hat{S}_1)$ **do**
  if $\sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{\hat{S}_1\cup\{j_1\}\setminus\{i_1\}, \hat{S}_1\cup\{j_1\}\setminus\{i_1\}})^\dagger} \boldsymbol{A}_{\hat{S}_1\cup\{j_1\}\setminus\{i_1\}, \hat{S}_2} \sqrt{(\boldsymbol{C}_{\hat{S}_2, \hat{S}_2})^\dagger}\right) > \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{\hat{S}_1, \hat{S}_1})^\dagger} \boldsymbol{A}_{\hat{S}_1, \hat{S}_2} \sqrt{(\boldsymbol{C}_{\hat{S}_2, \hat{S}_2})^\dagger}\right)$ **then**
    Update $\hat{S}_1 = \hat{S}_1 \cup \{j_1\} \setminus \{i_1\}$
  end if
end for
for each pair $(i_2, j_2) \in \hat{S}_2 \times ([m] \setminus \hat{S}_2)$ **do**
  if $\sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{\hat{S}_1\cup\{j_1\}\setminus\{i_1\}, \hat{S}_1\cup\{j_1\}\setminus\{i_1\}})^\dagger} \boldsymbol{A}_{\hat{S}_1\cup\{j_1\}\setminus\{i_1\}, \hat{S}_2} \sqrt{(\boldsymbol{C}_{\hat{S}_2, \hat{S}_2})^\dagger}\right) > \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{\hat{S}_1, \hat{S}_1})^\dagger} \boldsymbol{A}_{\hat{S}_1, \hat{S}_2} \sqrt{(\boldsymbol{C}_{\hat{S}_2, \hat{S}_2})^\dagger}\right)$ **then**
    Update $\hat{S}_2 = \hat{S}_2 \cup \{j_2\} \setminus \{i_2\}$
  end if
end for
while there is still an improvement
Output: $\hat{S}_1$, $\hat{S}_2$

C.1 Valid inequalities for SCCA

Before deriving the formulations, we first prove that there exists a bounded optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ of the SCCA. To be specific, we show that there exists an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ of the SCCA satisfying the constraints $\|\boldsymbol{x}^*\|_2^2 \le M_1$ and $\|\boldsymbol{y}^*\|_2^2 \le M_2$, where M_1 and M_2 are finite-valued parameters.

Proposition 3 *The SCCA admits an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ satisfying $\|\boldsymbol{x}^*\|_2^2 \le M_1$ and $\|\boldsymbol{y}^*\|_2^2 \le M_2$, where $M_1 = 1/\lambda_r(\boldsymbol{B}) + 1/(\lambda_r(\boldsymbol{B}) s_{\min}(\boldsymbol{B}))$ and $M_2 = 1/\lambda_{\hat{r}}(\boldsymbol{C}) + 1/(\lambda_{\hat{r}}(\boldsymbol{C}) s_{\min}(\boldsymbol{C}))$ with $\lambda_r(\boldsymbol{B}), \lambda_{\hat{r}}(\boldsymbol{C})$ being the smallest nonzero eigenvalues of matrices $\boldsymbol{B}, \boldsymbol{C}$ and $s_{\min}(\boldsymbol{R})$ being the smallest nonzero singular value of all the submatrices of the zero eigenvectors of matrix $\boldsymbol{R}$.*

Proof. Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ denote an optimal solution to SCCA. We bound $\|\boldsymbol{x}^*\|_2$ first and the same technique can be also straightforwardly applied to bound $\|\boldsymbol{y}^*\|_2$.

For matrix $\boldsymbol{B} \in \mathcal{S}_+^n$ of rank r, we let $\{\boldsymbol{q}_i\}_{i\in[n]} \in \mathbb{R}^n$ denote the eigenvectors corresponding to n eigenvalues $\boldsymbol{\lambda}$ of $\boldsymbol{B}$ such that $\lambda_1 \geq \ldots \geq \lambda_r > \lambda_{r+1} = \ldots = \lambda_n = 0$. Thus, $\{\boldsymbol{q}_i\}_{i\in[n]}$ are orthonormal and span the space of $\mathbb{R}^n$. Hence, there exists $\boldsymbol{\alpha} \in \mathbb{R}^n$ such that $\boldsymbol{x}^* = \sum_{i\in[n]} \alpha_i \boldsymbol{q}_i$. Given that $(\boldsymbol{x}^*)^\top \boldsymbol{B}\boldsymbol{x}^* \leq 1$, we have

$$\sum_{i\in[r]} \alpha_i^2 \lambda_i \leq 1.$$

Hence, the values of $\{\alpha_i\}_{i\in[r]}$ are bounded. On the other hand, let us define a subset $S \subseteq [n]$ of size at most s_1 such that $x_i^* \neq 0$ for each $i \in S$ and $x_j^* = 0$ for each $j \in [n] \setminus S$. Then for each $j \in [n] \setminus S$, we arrive at the following linear system:

$$\sum_{j\in[r+1,n]} \alpha_j \hat{\boldsymbol{q}}_j = -\sum_{i\in[r]} \alpha_i \hat{\boldsymbol{q}}_i, \tag{17}$$

where $\hat{\boldsymbol{q}}_i$ denote a subvector of $\boldsymbol{q}_i$ with indices $[n] \setminus S$ for each $i \in [n]$. For a fixed $\{\alpha_i\}_{i\in[r]}$, since the linear system (17) is nonempty, we let $\bar{\boldsymbol{Q}}\bar{\boldsymbol{\alpha}} = \bar{\boldsymbol{q}}$ denote its minimal linear subsystem such that a submatrix $\bar{\boldsymbol{Q}}$ is non-singular and the index set $\hat{S}$ of $\bar{\boldsymbol{\alpha}}$ is a subset of $[n] \setminus S$. Thus, we can construct an alternative solution $\hat{\boldsymbol{\alpha}}$ such that

$$\hat{\alpha}_i = \begin{cases} \alpha_i, & \text{if } i \in [r], \\ (\bar{\boldsymbol{Q}}^{-1}\bar{\boldsymbol{q}})_i, & \text{if } i \in \hat{S}, \\ 0, & \text{otherwise,} \end{cases}$$

and $\hat{\boldsymbol{x}} = \sum_{i\in[n]} \hat{\alpha}_i \boldsymbol{q}_i$. According to Lemma 1, we have

$$\hat{\boldsymbol{x}}^\top \boldsymbol{B}\hat{\boldsymbol{x}} \leq 1, \hat{\boldsymbol{x}}^\top \boldsymbol{A}\boldsymbol{y}^* = (\boldsymbol{x}^*)^\top \boldsymbol{A}\boldsymbol{y}^*,$$

i.e., $(\hat{\boldsymbol{x}}, \boldsymbol{y}^*)$ is also optimal to SCCA. Hence,

$$\|\hat{\boldsymbol{x}}\|_2 \leq \sqrt{\|\bar{\boldsymbol{Q}}^{-1}\bar{\boldsymbol{q}}\|_2^2 + \sum_{i\in[r]} \alpha_i^2}$$

Note that $\sum_{i\in[r]} \alpha_i^2 \leq 1/\lambda_r$ and

$$\|\bar{\boldsymbol{Q}}^{-1}\bar{\boldsymbol{q}}\|_2^2 \leq \|\bar{\boldsymbol{Q}}^{-1}\|_2^2 \|\bar{\boldsymbol{q}}\|_2^2 \leq \frac{1}{s_{\min}(\boldsymbol{B})} \frac{1}{\lambda_r}$$

where $s_{\min}(\boldsymbol{B})$ denotes the smallest nonzero singular values of all the submatrices of $[\boldsymbol{q}_{r+1}, \ldots, \boldsymbol{q}_n]$. In summary, we have

$$\|\hat{\boldsymbol{x}}\|_2 \leq \sqrt{1/\lambda_r + 1/(\lambda_r s_{\min}(\boldsymbol{B}))}.$$

This completes the proof. $\square$

The proof of Proposition 3 is straightforward in the case when $\boldsymbol{B}$ and $\boldsymbol{C}$ are of full rank as in this case the feasible region is a bounded set. In order to prove the result in the case when $\boldsymbol{B}$ is not full-rank, one has to show that it is possible to construct sparse solutions that are not "too far" away.

In fact, the bounds M_1, M_2 in Proposition 3 also hold for any given feasible subsets (S_1, S_2) of SCCA (1).

Corollary 2 *For any given feasible subsets (S_1, S_2) of SCCA 1, there exists a SCCA feasible solution $(\boldsymbol{x}, \boldsymbol{y})$ such that the supports of $\boldsymbol{x}, \boldsymbol{y}$ are S_1, S_2, respectively and we have that $\|\boldsymbol{x}\|_2^2 \leq M_1$ and $\|\boldsymbol{y}\|_2^2 \leq M_2$, where M_1, M_2 are defined in Proposition 3.*

C.2 Equivalent mixed-integer convex quadratic program of rank-one SCCA

When matrix $\boldsymbol{A}$ is rank-one, let us focus on analyzing the subproblem over $\boldsymbol{x}$ in (6), i.e.,

$$v_x = \max_{\boldsymbol{x}\in\mathbb{R}^n} \{\boldsymbol{a}^\top \boldsymbol{x} : \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \leq 1, \|\boldsymbol{x}\|_0 \leq s_1\}. \tag{18}$$

Then the second subproblem over $\boldsymbol{y}$ in (6) simply follows.

According to Corollary 2, introducing the binary variables $\boldsymbol{z}^1 \in \{0,1\}^n$ can reformulate the problem (18) as

$$v_x = \max_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{z}^1\in\{0,1\}^n} \left\{ \boldsymbol{a}^\top \boldsymbol{x} : \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x} \le 1, x_i \le \sqrt{M_1} z_i^1, \forall i \in [n], \sum_{i\in[n]} z_i^1 \le s_1 \right\}.$$

When matrix $\boldsymbol{B}$ is positive definite, there is a positive vector $\boldsymbol{b} \in \mathbb{R}^n_{++}$ and a positive semidefinite matrix $\hat{\boldsymbol{B}}$ such that $\boldsymbol{B} = \hat{\boldsymbol{B}} + \operatorname{Diag}(\boldsymbol{b})$. Given this equation, by leveraging the perspective techniques (see, e.g., [1, 43]), we can derive another equivalent MICQP formulation of the problem (18):

$$v_x = \max_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{z}^1\in\{0,1\}^n, \boldsymbol{\mu}\in\mathbb{R}^n_+} \left\{ \boldsymbol{a}^\top \boldsymbol{x} : \boldsymbol{x}^\top \hat{\boldsymbol{B}}\boldsymbol{x} + \sum_{i\in[n]} \mu_i \le 1, x_i^2 \le \mu_i z_i^1, \forall i \in [n], \sum_{i\in[n]} z_i^1 \le s_1 \right\}.$$

which is often stronger than the above formulation.

D A branch-and-cut algorithm with closed-form cuts

By dualizing the inner maximization problem over $\boldsymbol{X}$ in the MISDP (7), in this subsection, we derive an equivalent mixed-integer linear program for SCCA, which motivates us to develop a branch-and-cut algorithm.

By introducing the Lagrangian multipliers $(\theta_1, \theta_2, \boldsymbol{\lambda})$, the Lagrangian dual of the maximization problem (9) can be written as

$$\begin{aligned} f(\boldsymbol{z}) = & \min_{\substack{\theta_1\ge 0, \theta_2 \ge 0, \\ \boldsymbol{\lambda}\in\mathbb{R}^{n+m}_+}} \max_{\boldsymbol{X}\in\mathcal{S}^{n+m}_+} \operatorname{tr}(\tilde{\boldsymbol{A}}\boldsymbol{X}) - \theta_1 \operatorname{tr}(\tilde{\boldsymbol{B}}\boldsymbol{X}) - \theta_2 \operatorname{tr}(\tilde{\boldsymbol{C}}\boldsymbol{X}) + \theta_1 + \theta_2, \\ & - \sum_{i\in[n+m]} \lambda_i X_{ii} + \sum_{i\in[n+m]} \lambda_i M_{ii} z_i \\ = & \min_{\substack{\theta_1\ge 0, \theta_2 \ge 0, \\ \boldsymbol{\lambda}\in\mathbb{R}^{n+m}_+}} \left\{ \theta_1 + \theta_2 + \sum_{i\in[n+m]} \lambda_i M_{ii} z_i : \begin{pmatrix} \theta_1 \boldsymbol{B} & -\boldsymbol{A}/2 \\ -\boldsymbol{A}^\top/2 & \theta_2 \boldsymbol{C} \end{pmatrix} \succeq -\operatorname{Diag}(\boldsymbol{\lambda}) \right\}, \end{aligned} \tag{19}$$

where the strong duality holds due to the function $f(\boldsymbol{z})$ being concave, bounded, and thus continuous in the set $\hat{\mathcal{Z}}$ and Slater condition holds for any interior point $\boldsymbol{z}$ in the set $\hat{\mathcal{Z}}$.

Below, we derive the closed-form expression of the function $f(\boldsymbol{z})$ with the given binary variable $\boldsymbol{z} \in \mathcal{Z}$. This allows us to reformulate SCCA (8) as a mixed-integer linear program with exponentially many linear constraints and an efficient separation oracle.

Proposition 4 *The SCCA (8) is equivalent to*

$$\begin{aligned} v^* = \max_{\boldsymbol{z}\in\mathcal{Z}, v} \Bigg\{ v : v \le & \ \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger} \boldsymbol{A}_{S_1,S_2} \sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger} \right) + \\ & \sum_{i\in S_1\cup(S_2+n)} \lambda^* M_{ii} z_i : \forall S_1 \subseteq [n], |S_1| \le s_1, S_2 \subseteq [m], |S_2| \le s_2 \Bigg\}, \end{aligned} \tag{20}$$

where for a pair of subsets (S_1, S_2), the scalar λ^* is defined as the largest positive eigenvalue of matrix $\boldsymbol{D}_2^\top \boldsymbol{D}_1^{-1} \boldsymbol{D}_2 - \boldsymbol{D}_3$ with

$$\boldsymbol{D}_1 = \begin{pmatrix} \theta_1^* \boldsymbol{B}_{S_1,S_1} & -\boldsymbol{A}_{S_1,S_2}/2 \\ -\boldsymbol{A}_{S_1,S_2}^\top/2 & \theta_2^* \boldsymbol{C}_{S_2,S_2} \end{pmatrix}, \quad \boldsymbol{D}_2 = \begin{pmatrix} \theta_1^* \boldsymbol{B}_{S_1,[n]\backslash S_1} & -\boldsymbol{A}_{S_1,[m]\backslash S_2}/2 \\ -\boldsymbol{A}_{S_2,[n]\backslash S_1}^\top/2 & \theta_2^* \boldsymbol{C}_{S_2,[m]\backslash S_2} \end{pmatrix},$$

and

$$\boldsymbol{D}_3 = \begin{pmatrix} \theta_1^* \boldsymbol{B}_{[n]\backslash S_1,[n]\backslash S_1} & -\boldsymbol{A}_{[n]\backslash S_1,[m]\backslash S_2}/2 \\ -\boldsymbol{A}_{[n]\backslash S_1,[m]\backslash S_2}^\top/2 & \theta_2^* \boldsymbol{C}_{[m]\backslash S_2,[m]\backslash S_2} \end{pmatrix},$$

where $\theta_1^* = \theta_2^* = \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger} \boldsymbol{A}_{S_1,S_2} \sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger} \right)/2$.

Proof. First, for any binary variable $\boldsymbol{z} \in \mathcal{Z}$, suppose $S_1 = \{i : z_i = 1, \forall i \in [n]\}$, $S_2 = \{i - n : z_i = 1, \forall i \in [n+1, n+m]\}$, and $T \subseteq [n+m]$ denotes the support of $\boldsymbol{z}$. Then following the proof of Proposition 1, we can construct a rank-one optimal solution $\boldsymbol{X}^* = \begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^* \\ \boldsymbol{y}^* \end{pmatrix}^\top$ to the maximization problem below that admits the optimal value $\sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger}\boldsymbol{A}_{S_1,S_2}\sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger}\right)$, i.e.,

$$\begin{aligned} &\max_{\boldsymbol{X} \in \mathcal{S}_+^{n+m}} \{\operatorname{tr}(\tilde{\boldsymbol{A}}\boldsymbol{X}) : \operatorname{tr}(\tilde{\boldsymbol{B}}\boldsymbol{X}) \le 1, \operatorname{tr}(\tilde{\boldsymbol{C}}\boldsymbol{X}) \le 1, X_{ii} = 0, \forall i \in [n+m] \setminus T\} \\ &= \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger}\boldsymbol{A}_{S_1,S_2}\sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger}\right) \ge f(\boldsymbol{z}), \end{aligned}$$

where the inequality is because the maximization problem above relaxes the valid constraints $X_{ii} \le M_{ii}$ for all $i \in T$ in maximization problem (9). The result in Corollary 2 suggests that $\boldsymbol{x}^*, \boldsymbol{y}^*$ can be bounded and their two norms must not exceed M_1, M_2, which means that the optimal solution $\boldsymbol{X}^*$ satisfies the $X_{ii} \le M_{ii}$ for all $i \in T$. Therefore, $\boldsymbol{X}^*$ is feasible and optimal to maximization problem (9) and we have that

$$f(\boldsymbol{z}) = \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger}\boldsymbol{A}_{S_1,S_2}\sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger}\right).$$

According to strong duality, the minimization problem (19) admits an optimal value $\sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger}\boldsymbol{A}_{S_1,S_2}\sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger}\right)$. Next, we construct its optimal solution $(\theta_1^*, \theta_2^*, \boldsymbol{\lambda}^*)$.

For any given $\epsilon > 0$, we let $\theta_1^* = f(\boldsymbol{z})/2$, $\theta_2^* = f(\boldsymbol{z})/2$, $\hat{\lambda}_i(\epsilon) = \frac{\epsilon}{M_{ii}|T|}$ for all $i \in T$, and $\hat{\lambda}_i(\epsilon) = \lambda^*(\epsilon)$ for all $i \in [n] \setminus T$, where

$$\lambda^*(\epsilon) = \left[\lambda_{\max}\left(\boldsymbol{D}_2^\top\left(\boldsymbol{D}_1 + \operatorname{Diag}\left(\hat{\boldsymbol{\lambda}}_T(\epsilon)\right)\right)^{-1}\boldsymbol{D}_2 - \boldsymbol{D}_3\right)\right]_+.$$

It is easy to compute that $\theta_1^* + \theta_2^* + \sum_{i\in[n+m]}\hat{\lambda}_i(\epsilon)M_{ii}z_i = f(\boldsymbol{z}) + \epsilon$. Thus, for any $\epsilon > 0$, if $(\theta_1^*, \theta_2^*, \hat{\boldsymbol{\lambda}}(\epsilon))$ were feasible, then it is an ϵ-optimal solution to the minimization problem (19). It remains to verify the feasibility of the solution $(\theta_1^*, \theta_2^*, \hat{\boldsymbol{\lambda}}(\epsilon))$, i.e., checking the constraint below

$$\begin{pmatrix} \theta_1^*\boldsymbol{B} & -\boldsymbol{A}/2 \\ -\boldsymbol{A}^\top/2 & \theta_2^*\boldsymbol{C} \end{pmatrix} + \operatorname{Diag}\left(\hat{\boldsymbol{\lambda}}(\epsilon)\right) \succeq 0.$$

By performing the permutation of the rows and columns of the above matrix, it is sufficient to show that the new block matrix

$$\begin{pmatrix} \boldsymbol{D}_1 + \operatorname{Diag}\left(\hat{\boldsymbol{\lambda}}_T(\epsilon)\right) & \boldsymbol{D}_2 \\ \boldsymbol{D}_2^\top & \boldsymbol{D}_3 + \lambda^*(\epsilon)\boldsymbol{I} \end{pmatrix} \succeq 0, \tag{21}$$

is positive semidefinite.

Since $\begin{pmatrix} \boldsymbol{B}_{S_1,S_1} & -\boldsymbol{A}_{S_1,S_2}/2 \\ -\boldsymbol{A}_{S_1,S_2}^\top/2 & \boldsymbol{C}_{S_2,S_2} \end{pmatrix}$ is a principal submatrix of a positive semidefinite matrix $\begin{pmatrix} \boldsymbol{B} & -\boldsymbol{A}/2 \\ -\boldsymbol{A}^\top/2 & \boldsymbol{C} \end{pmatrix}$, it is also positive semidefinite. According to Lemma 1 and the fact that $\theta_1^* = \theta_2^* = \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger}\boldsymbol{A}_{S_1,S_2}\sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger}\right)/2$, the matrix $\boldsymbol{D}_1$ is also positive semidefinite. As $\epsilon > 0$, the matrix $\boldsymbol{D}_1 + \operatorname{Diag}\left(\hat{\boldsymbol{\lambda}}_T(\epsilon)\right)$ must be positive definite, which means that

$$\left(\boldsymbol{I} - \left(\boldsymbol{D}_1 + \operatorname{Diag}\left(\hat{\boldsymbol{\lambda}}_T(\epsilon)\right)\right)\left(\boldsymbol{D}_1 + \operatorname{Diag}\left(\hat{\boldsymbol{\lambda}}_T(\epsilon)\right)\right)^{-1}\right)\boldsymbol{D}_2 = \boldsymbol{0}.$$

Besides, according to the definition of $\lambda^*(\epsilon)$, we obtain

$$\boldsymbol{D}_3 + \lambda^*(\epsilon)\boldsymbol{I} - \boldsymbol{D}_2^\top\left(\boldsymbol{D}_1 + \operatorname{Diag}\left(\hat{\boldsymbol{\lambda}}_T(\epsilon)\right)\right)^{-1}\boldsymbol{D}_2 \succeq \boldsymbol{0}.$$

Taking these results together, according to Lemma 1, the constraint in (21) must hold for a given solution $(\theta_1^*, \theta_2^*, \hat{\boldsymbol{\lambda}}(\epsilon))$. Since the objective value corresponding to $(\theta_1^*, \theta_2^*, \hat{\boldsymbol{\lambda}}(\epsilon))$ is at most ϵ larger than the optimal value of problem (19), letting $\epsilon \to 0$ and using the closedness of the feasible set in problem (19), we can confirm the optimality of $(\theta_1^*, \theta_2^*, \boldsymbol{\lambda}^*)$ with $\lambda_i^* = 0$ for all $i \in T$ and $\lambda_i^* = \lambda^*$ for all $i \in [n] \setminus T$.

Given the closed-form optimal solution to problem (19), the rest of the proof follows from [28, theorem 7]. □

We note that SCCA (20) can be implemented via a delayed cut-generation procedure. That is, at each feasible branch-and-bound node with a binary solution $\hat{\boldsymbol{z}}$, let $S_1 = \{i : \hat{z}_i = 1, \forall i \in [n]\}$ and $S_2 = \{i - n : \hat{z}_i = 1, \forall i \in [n+1, n+m]\}$. Then we can compute the corresponding scalar λ^* and generate the following valid inequality based on (20):

$$v \le \sigma_{\max}\left(\sqrt{(\boldsymbol{B}_{S_1,S_1})^\dagger}\boldsymbol{A}_{S_1,S_2}\sqrt{(\boldsymbol{C}_{S_2,S_2})^\dagger}\right) + \sum_{i \in S_1 \cup (S_2+n)} \lambda^* M_{ii} z_i.$$

E Data description

Table 7: Description of UCI and breast cancer datasets used

Dataset	# of variables	# of samples	n	m	rank r	rank $\hat{r}$
dermatology	34	366	17	17	17	17
spambase	57	4601	28	29	28	29
digits	64	1797	32	32	32	32
buzz	77	583250	38	39	38	39
gas	128	2565	64	64	64	64
slice	385	53500	192	193	192	193
breast cancer	21821	89	19,672	2,149	89	89

F Multiple Sparse Canonical Correlation Analysis

The multiple CCA problem can be formulated as follows:

$$\max_{\boldsymbol{x} \in \mathbb{R}^{n \times k}, \boldsymbol{y} \in \mathbb{R}^{m \times k}} \left\{ \mathrm{tr}(\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}) : \boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{x} = \boldsymbol{I}_k, \boldsymbol{y}^\top \boldsymbol{C} \boldsymbol{y} = \boldsymbol{I}_k \right\},$$

where k denotes the number of pairs of basis vectors and $\boldsymbol{I}_k$ denotes the identity matrix of size k.

As $\boldsymbol{x}, \boldsymbol{y}$ can be matrices, we propose adding row sparse constraints to extend SCCA for multiple vectors, which is defined as:

$$\max_{\boldsymbol{x} \in \mathbb{R}^{n \times k}, \boldsymbol{y} \in \mathbb{R}^{m \times k}} \left\{ \mathrm{tr}(\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}) : \boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{x} = \boldsymbol{I}_k, \boldsymbol{y}^\top \boldsymbol{C} \boldsymbol{y} = \boldsymbol{I}_k, \|\boldsymbol{x}\|_0 \le s_1, \|\boldsymbol{y}\|_0 \le s_2 \right\},$$

where we let $\|\boldsymbol{x}\|_0$ and $\|\boldsymbol{y}\|_0$ denote the number of nonzero rows of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

This multiple SCCA model can (i) compute the multiple weight vectors $(\boldsymbol{x}, \boldsymbol{y})$ simultaneously and (ii) enforce the sparsity and orthogonality strictly. To be specific, the constraints $\boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{x} = \boldsymbol{I}_k, \boldsymbol{y}^\top \boldsymbol{C} \boldsymbol{y} = \boldsymbol{I}_k$ ensure the orthogonal left- and right-canonical loading vectors in multiple SCCA. By the definition of row sparsity, the resultant multiple left- and right-basis vectors, i.e., the columns of $\boldsymbol{x}$ and $\boldsymbol{y}$, share the same nonzero rows, respectively.

More importantly, the row-sparsity enables us to readily extend the proposed algorithms to solve multiple SCCA. We have tested them on UCI data, and the computational results are presented in Table 8. As k increases, it takes branch-and-cut a longer time to return an optimal solution.

Table 8: Evaluation of our algorithms for solving multiple SCCA on UCI datasets

					Greedy		Local search		Convex relaxation			Branch-and-cut		
n	m	s_1	s_2	k	LB	time	LB	time	UB	gap(%)	time	v^*	MIPGap(%)	time
17	17	5	5	2	1.907	0.01	1.935	0.06	1.957	1.14	0.01	1.935	0.00	2
17	17	10	10	3	2.879	0.02	2.884	0.09	2.898	0.45	0.01	2.884	0.00	6
28	29	5	5	2	1.182	0.02	1.233	0.09	1.358	10.19	0.01	1.233	0.00	234
28	29	10	10	3	1.579	0.04	1.586	0.14	1.685	6.23	0.01	1.587	5.33	–
32	32	5	5	2	1.906	0.02	1.906	0.04	1.935	1.55	0.01	1.916	0.00	14
32	32	10	10	3	2.736	0.04	2.741	0.19	2.770	1.05	0.01	2.742	0.00	3093
38	39	5	5	2	2	0.03	2	0.25	2	0.00	0.01	2	0.00	8
38	39	10	10	3	3	0.05	3	0.59	3	0.00	0.01	3	0.00	10
64	64	5	5	2	1.947	0.05	1.991	0.34	1.997	0.29	0.01	1.993	0.15	–
64	64	10	10	3	2.983	0.09	2.989	0.71	2.993	0.14	0.02	2.989	0.14	–
192	193	5	5	2	1.911	0.21	1.991	2.36	1.995	0.22	0.04	1.991	0.21	–
192	193	10	10	3	2.907	0.38	2.951	5.78	2.977	0.90	0.04	2.954	0.78	–

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and/or introduction should clearly state the claims made, including the629 contributions made in the paper and important assumptions and limitation

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the proposed algorithms when analyzing the numerical results of each table in Section 5. The paper points out the preconditions to apply Algorithm 1 and to reduce SCCA to MICQPs in Subsection 3.2.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The proofs can either appear in the main paper or the supplemental material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 discusses the generation of experimental results. The supplementary material also contains necessary codes and data used for result reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The supplementary material contains necessary codes and data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the experimental setting in Subsection 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper focus on solving an optimization problem with deterministic data and evaluates the optimality performance of the algorithms. Hence, the paper does not include error bars or other statistical significance measures of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the details on the compute resources in the beginning of Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and make sure that we have conformed with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper improves the interpretability of the widely-used CCA method, making it more convincing when applied to real data analytics, e.g., genetic analytics of breast cancer data. We do not foresee any negative societal impact of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original paper that produced the code package or dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper does not release new assets

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.