Exact and Approximation Algorithms for Sparse PCA

Yongchun Li
Department of Industrial & Systems Engineering, Virginia Tech, Blacksburg, VA 24061, liyc@vt.edu

Weijun Xie
Department of Industrial & Systems Engineering, Virginia Tech, Blacksburg, VA 24061, wxie@vt.edu

Sparse PCA (SPCA) is a fundamental model in machine learning and data analytics, which has witnessed a variety of application areas such as finance, manufacturing, biology, healthcare. To select a prespecified-size principal submatrix from a covariance matrix to maximize its largest eigenvalue for the better interpretability purpose, SPCA advances the conventional PCA with both feature selection and dimensionality reduction. Existing approaches often approximate SPCA as a semi-definite program (SDP) without strictly enforcing the important cardinality constraint that restricts the number of selected features to be a constant. To fill this gap, we propose two exact mixed-integer SDPs (MISDPs) by exploiting the spectral decomposition of the covariance matrix and the properties of the largest eigenvalues. We then analyze the theoretical optimality gaps of their continuous relaxation values and prove that they are stronger than that of the state-of-art one. We further show that the continuous relaxations of two MISDPs can be recast as saddle point problems without involving semi-definite cones, and thus can be effectively solved by first-order methods such as the subgradient method. Since off-the-shelf solvers, in general, have difficulty in solving MISDPs, we approximate SPCA with arbitrary accuracy by a mixed-integer linear program (MILP) of a similar size as MISDPs. The continuous relaxation values of two MISDPs can be leveraged to reduce the size of the proposed MILP further. To be more scalable, we also analyze greedy and local search algorithms, prove their first-known approximation ratios, and show that the approximation ratios are tight. Our numerical study demonstrates that the continuous relaxation values of the proposed MISDPs are quite close to optimality, the proposed MILP model can solve small and medium-size instances to optimality, and the approximation algorithms work very well for all the instances. Finally, we extend the analyses to Rank-one Sparse SVD (R1-SSVD) with non-symmetric matrices and Sparse Fair PCA (SFPCA) when there are multiple covariance matrices, each corresponding to a protected group.

Key words: Sparse PCA, Largest Eigenvalue, Mixed-Integer Program, Semi-definite Program, Greedy, Local Search, SVD, Fairness

---

1. Introduction This paper studies the sparse principal component analysis (SPCA) problem of the form

$$\text{(SPCA)} \quad w^* := \max_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} : ||\boldsymbol{x}||_2 = 1, ||\boldsymbol{x}||_0 = k \right\}, \tag{1}$$

where the symmetric positive semi-definite matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ denotes the sample covariance out of a dataset with n features and the integer $k \in [n]$ denotes the sparsity of its first principal component (PC). In SPCA (1), the objective is to select the best size-k principal submatrix from a covariance matrix $\boldsymbol{A}$ with the maximum largest eigenvalue. Compared to the conventional PCA, the extra zero-norm constraint $||\boldsymbol{x}||_0 = k$ in SPCA (1) restricts the number of features of the first PC $\boldsymbol{x}$ to be k most important ones. In this way, SPCA improves the interpretability of the obtained PC, which has been shown as early as Jeffers [20] in 1967. It is also recognized that SPCA can be more reliable for large-scale datasets than PCA, where the number of features is far more than that of observations [41]. These advantages of SPCA have benefited many application fields such as biology, finance, cloud computing, and healthcare, which frequently deal with datasets with a massive number of features (see, e.g., [8, 21, 25, 30]).

1.1. Relevant Literature Our paper contributes to relevant literature on SPCA from three aspects: exact mixed-integer programs, convex relaxations, and approximation algorithms.

Exact Mixed-Integer Programs: As shown in formulation (1), SPCA is highly non-convex-maximizing a convex function subject to two nonconvex constraints (i.e., an L_2 equality constraint and an L_0 equality constraint). Albeit superior to traditional PCA, SPCA (1) is notoriously known to be computationally expensive; see, e.g., the complexity analysis and inapproximability results in Magdon-Ismail [27]. As a result, the equivalent formulations and algorithms for exactly solving SPCA are quite limited in the literature (see, e.g., [5, 17, 29]). Moghaddam et al. [29] introduced a branch and bound method to solve SPCA, and they pruned redundant nodes using the eigenvalue of principal submatrices and a greedy algorithm. Recently, Berk and Bertsimas [5] embedded various upper and lower bounds into this branch and bound framework, which could efficiently prune nodes and quickly certificate the optimality for quite a few instances. It is worthy of mentioning that Gally and Pfetsch [17] proposed a MISDP (MISDP) formulation for SPCA. Our second MISDP formulation differs from Gally and Pfetsch [17] by deriving two strong conic valid inequalities. Another interesting work can be found in Dey et al. [15], where the authors developed approximate convex integer programs for SPCA with an optimality gap of $(1+\sqrt{k/(k+1)})^2$. Quite differently, we propose two exact MISDP formulations and one approximate mixed-integer linear program (MILP) for SPCA from novel perspectives of analyzing the largest eigenvalue. Specifically, the proposed MILP formulation can be arbitrarily close to the optimal value of SPCA, and it can be directly solved by off-the-shelf solvers such as Gurobi.

Convex Relaxations: Besides solving exact SPCA, researchers have also actively sought to explore effective convex relaxations. A common approach in literature is to develop SDP relaxations

for SPCA (see e.g., [1, 13, 16, 12, 40]). Albeit convex, solvers often have difficulty in solving large-scale instances of SDP formulations (e.g., $n = \Omega(100)$). The computational challenge of these SDP problems urgently calls for more effective methods to compute the relaxation values for SPCA. From a different angle, this paper solves the continuous relaxations of the proposed MISDP formulations as the maximin saddle point problem, where the subgradient method enjoys a $O(1/T)$ rate of convergence [31] based on Euclidean projections. Surprisingly, we further show that the projection oracle of the subgradient method is a second-order conic program rather than an SDP and thus can be easily dealt with.

Approximation Algorithm: Another early thread of research on SPCA is the development of high-quality heuristics for solving SPCA to near optimality such as greedy algorithm [16, 19], truncation algorithm [9], power method [22], and variable neighborhood search method [7]. In particular, the truncation algorithm in [9] so far provides the best-known approximation ratio $O(n^{-1/3})$, which can be easily implemented to generate a feasible solution for SPCA. This paper investigates the greedy and local search algorithms and proves their first-known approximation ratios $O(1/k)$ for SPCA.

1.2. Summary of Contributions We observe that when the support of $\boldsymbol{x}$ has been successfully identified, SPCA (1) reduces to the conventional PCA finding the largest eigenvalue and eigenvector of a size-k principal submatrix of $\boldsymbol{A}$. This fact motivates us to derive two equivalent MISDP formulations and an approximate MILP of SPCA. Below is a summary of the main contributions in this paper.

(i) For each formulation, we derive the theoretical optimality gap between its continuous relaxation value and the optimal value of SPCA.

(ii) Our first MISDP formulation inspires us to derive closed-form expressions of the coefficients of valid inequalities, which can be efficiently embedded into the branch and cut algorithms;

(iii) We show that the subgradient method can be adapted to ease the computational burden of obtaining MISDP continuous relaxation values with $O(1/T)$ rate of convergence. These continuous relaxations values can further help reduce the size of MILP;

(iv) The continuous relaxation of our second MISDP formulation is proven to be stronger than the one proposed in d'Aspremont et al. [13];

(v) The proposed MILP formulation has a similar size as two MISDPs and can be directly solved using many existing solvers;

(vi) We prove and demonstrate the tightness of the first-known approximation ratios for the greedy and local search algorithms;

(vii) Our analyses can be extended to the Rank-one Sparse SVD (R1-SSVD), which aims to compute the largest singular value of the possibly non-symmetric matrix $\boldsymbol{A}$ with the sparsity constraints on its left-singular and right-singular vectors separately; and

(viii) We extend the second MISDP formulation to Sparse Fair PCA (SFPCA), where the covariance matrices are observed from multiple protected groups.

Our contributions have both theoretical and practical relevance. Theoretically, we contribute three exact mixed-integer convex programs to SPCA. Practically, our MILP formulation can either attain optimal solutions for SPCA, improve the continuous relaxations, or find better-quality feasible solutions for small and medium-size instances. We apply the computationally efficient subgradient method to solving the continuous relaxations of the proposed MISDPs, as well as deriving their theoretical optimality gaps. We also develop two scalable approximation algorithms to solve SPCA to near optimality and prove their approximation ratios. Our proposed algorithms have been demonstrated to be successfully applied to large-scale data analytics problems, such as identifying key features for the drug abuse problem. We further extend the analyses to R1-SSVD and SFPCA. All the theoretical contributions are summarized in Table 1.

TABLE 1. Summary of Theoretical Contributions

Problem	**Exact Mixed Integer Program**	**Optimality Gap[2]**
SPCA	MISDP (6)	$\min\{k, nk^{-1}\}$
	MISDP (15)	$k, nk^{-1}\}$
	MILP (22)	$\min\{k(\sqrt{d}/2+1/2), nk^{-1}\sqrt{d}+(n-k)(\sqrt{d}/2+1/2)\}$
R1-SSVD	MISDP (34)	$\sqrt{mnk_1^{-1}k_2^{-1}}$
	MISDP (35)	$\min\{\sqrt{k_1k_2}, \sqrt{mnk_1^{-1}k_2^{-1}}\}$
	MILP (36)	$\sqrt{mnk_1^{-1}k_2^{-1}}[\min\{(k_1+k_2)(\sqrt{d}/2+1/2),$ $mnk_1^{-1}k_2^{-1}\sqrt{d}+(m+n-k_1-k_2)(\sqrt{d}/2+1/2)\}-1]$
SFPCA[2]	MISDP (40)	–

Problem	**Approximation Algorithm**	**Approximation Ratio[3]**
SPCA	Greedy Algorithm 1	k^{-1}
	Local Search Algorithm 2	k^{-1}
R1-SSVD	Truncation algorithm	$\max\{\sqrt{k_1^{-1}}, \sqrt{k_2^{-1}}, \sqrt{k_1k_2m^{-1}n^{-1}}\}$
	Greedy Algorithm 3	$\sqrt{k_1^{-1}k_2^{-1}}$
	Local Search Algorithm 4	$\sqrt{k_1^{-1}k_2^{-1}}$

[1] Optimality Gap is the ratio between the continuous relaxation value and the optimal one;
[2] The formulation (40) provides an upper bound for general SFPCA and becomes exact when there are only two groups;
[3] Approximation Ratio denotes the ratio between the objective value of an approximation algorithm and the optimal one.

Organization: The remainder of this paper is organized as follows. Sections 2 and 3 develop two MISDP formulations for SPCA and prove the optimality gaps of their continuous relaxation values. Section 4 investigates an approximate MILP, which can be arbitrarily close to the optimal value of SPCA, and proves the optimality gap of its continuous relaxation value. Section 5 introduces and analyzes two approximation algorithms. Section 6 conducts a numerical study to demonstrate the efficiency and the solution quality of our proposed formulations and algorithms. Sections 7 and 8 separately extend the analyses to the rank-one sparse SVD (R1-SSVD) and the sparse fair PCA (SFPCA). Finally, conclusion and future directions are exhibited in Section 9.

Notation: The following notation is used throughout the paper. We let $\mathcal{S}^n, \mathcal{S}^n_+, \mathcal{S}^n_{++}$ denote set of all the $n\times n$ symmetric real matrices, set of all the $n\times n$ symmetric positive semi-definite matrices, and set of all the $n\times n$ symmetric positive definite matrices, respectively. 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 use corresponding non-bold letters (e.g., x_i, X_{ij}) to denote their components. We use $\mathbf{0}$ to denote the zero vector and $\mathbf{1}$ to denote the all-ones vector. We use $\lceil\cdot\rceil$ as a ceil function. We let $\mathbb{R}^n_+$ denote the set of all the n dimensional nonnegative vectors and let $\mathbb{R}^n_{++}$ denote the set of all the n dimensional positive vectors. Given a positive integer n and an integer $s\leq n$, we let $[n]:=\{1,2,\cdots,n\}$ and let $[s,n]:=\{s,s+1,\cdots,n\}$. We let $\boldsymbol{I}_n$ denote the $n\times n$ identity matrix and let $\boldsymbol{e}_i$ denote its i-th column vector. Given a set S and an integer k, we let $|S|$ denote its cardinality and $\binom{S}{k}$ denote the collection of all the size-k subsets out of S. Given an $m\times n$ matrix $\boldsymbol{A}$ and two sets $S\in[m]$, $T\in[n]$, we let $\boldsymbol{A}_{S,T}$ denote a submatrix of $\boldsymbol{A}$ with rows and columns indexed by sets S,T, respectively and let $\boldsymbol{A}_S$ denote a submatrix of $\boldsymbol{A}$ with columns from the set S. Given a vector $\boldsymbol{x}\in\mathbb{R}^n$, we let $\mathrm{Diag}(\boldsymbol{x})$ denote the diagonal matrix with diagonal elements $x_1,\cdots,x_n$, and let $\mathrm{supp}(\boldsymbol{x})$ denote the support of $\boldsymbol{x}$. Given a square symmetric matrix $\boldsymbol{A}$, let $\mathrm{diag}(\boldsymbol{A})$ denote the vector of diagonal entries of $\boldsymbol{A}$, and let $\lambda_{\min}(\boldsymbol{A}), \lambda_{\max}(\boldsymbol{A})$ denote the smallest and largest eigenvalues of $\boldsymbol{A}$, respectively. Given a non-square matrix $\boldsymbol{A}$, let $\sigma_{\max}(\boldsymbol{A})$ denote the largest singular value. Additional notation will be introduced later as needed.

2. Exact MISDP Formulation (I) In this section, we derive an equivalent mixed-integer semi-definite programming (MISDP) formulation for SPCA based on the spectral decomposition and disjunctive programming techniques.

To begin with, for each $i\in[n]$, we let the binary variable $z_i=1$ if the i-th feature is selected, and 0, otherwise. Linearizing the zero-norm constraint using binary vector $\boldsymbol{z}$, then SPCA (1) can be equivalently formulated as a following nonconvex mixed-integer quadratic program:

$$\text{(SPCA)}\quad w^*:=\max_{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{z}\in Z}\left\{\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}:||\boldsymbol{x}||_2=1,|x_i|\leq z_i,\forall i\in[n]\right\},\tag{2}$$

where we let cardinality set Z denote the feasible region of $\boldsymbol{z}$, i.e.,

$$Z = \left\{ \boldsymbol{z} \in \{0,1\}^n : \sum_{i\in[n]} z_i = k \right\}.$$

For SPCA (2), we note that (i) the binary vector $\boldsymbol{z}$ is of vital importance and its associated feasible region Z will be used throughout this paper for two MISDPs and one MILP, and (ii) the derivations of all the three mixed-integer formulations originate from the naive SPCA (2).

2.1. Spectral Reformulation We observe that given a size-k subset of features (i.e., the support of the binary vector $\boldsymbol{z}$ in formulation (2) is specified), the SPCA (2) is equivalent to finding the largest eigenvalue of the corresponding principal submatrix of $\boldsymbol{A}$. This fact inspires us to propose three equivalent mixed-integer convex programs for SPCA (2) . This observation is summarized below.

Lemma 1 *For a symmetric matrix $\boldsymbol{A} \in \mathcal{S}^n$ and a size-k set $S \subseteq [n]$, the followings must hold:*

(i) $\max_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} : ||\boldsymbol{x}||_2 = 1, x_i = 0, \forall i \notin S \right\} = \lambda_{\max}(\boldsymbol{A}_{S,S})$,

(ii) $\max_{\boldsymbol{X}\in\mathcal{S}_+^k} \{\mathrm{tr}(\boldsymbol{A}_{S,S}\boldsymbol{X}) : \mathrm{tr}(\boldsymbol{X}) = 1\} = \lambda_{\max}(\boldsymbol{A}_{S,S})$, *and*

(iii) If matrix $\boldsymbol{A}$ is positive semi-definite, then $\lambda_{\max}(\boldsymbol{A}_{S,S}) = \lambda_{\max}(\sum_{i\in S} \boldsymbol{c}_i \boldsymbol{c}_i^\top)$, where $\boldsymbol{A} = \boldsymbol{C}^\top \boldsymbol{C}$, $\boldsymbol{C} \in \mathbb{R}^{d\times n}$ denotes the Cholesky factorization matrix of $\boldsymbol{A}$, d is the rank of $\boldsymbol{A}$, and $\boldsymbol{c}_i \in \mathbb{R}^d$ denotes i-th column vector of $\boldsymbol{C}$ for each $i \in [n]$.

Proof. See Appendix A.1. □

The results in Lemma 1 are crucial to this paper and allow us to derive the exact mixed-integer convex programs of SPCA. Specifically, we remark that: Part (i) of Lemma 1 reduces SPCA to selecting the best size-k principal submatrix of $\boldsymbol{A}$ to achieve the maximum largest eigenvalue, which establishes a combinatorial formulation of SPCA; Part (ii) of Lemma 1 shows that SDP relaxation of the largest eigenvalue problem by dropping the rank-one constraint is exact and inspires us to develop two MISDP formulations for SPCA; and since the covariance matrix used in SPCA is always positive semi-definite, the identity in Part (iii) of Lemma 1 suggests an alternative way of formulating SPCA using Cholesky decomposition, which motivates us to derive an exact MISDP formulation in this section and an MILP in a later section.

According to Part (i) in Lemma 1, introducing a subset S, a natural combinatorial reformulation of SPCA (1) is defined as:

$$w^* := \max_S \{\lambda_{\max}(\boldsymbol{A}_{S,S}) : |S| = k, S \subseteq [n]\}. \tag{3}$$

By computing the Cholesky factorization of $\boldsymbol{A}=\boldsymbol{C}^\top\boldsymbol{C}$ with $\boldsymbol{C}\in\mathbb{R}^{d\times n}$ and d denoting the rank of $\boldsymbol{A}$, then the identity in Part (iii) in Lemma 1 recasts the objective function of SPCA (3) as below:

$$w^*:=\max_{S}\left\{\lambda_{\max}\left(\sum_{i\in S}\boldsymbol{c}_i\boldsymbol{c}_i^\top\right):|S|=k,S\subseteq[n]\right\}. \tag{4}$$

Recall that for each $i\in[n]$, binary variable $z_i=1$ if ith feature (i.e., column $\boldsymbol{c}_i$) is selected, and 0, otherwise. Therefore, SPCA (4) can be further reformulated as

$$w^*:=\max_{\boldsymbol{z}\in Z}\left\{\lambda_{\max}\left(\sum_{i\in[n]}z_i\boldsymbol{c}_i\boldsymbol{c}_i^\top\right)\right\}. \tag{5}$$

The above formulation involves with concave objective function but it is a maximization problem, which will cause much trouble. Fortunately, the result in Part (ii) of Lemma 1 and the reformulation technique from disjunctive programming [2] motivate us to convert SPCA (5) to an equivalent MISDP, which is shown as below.

Theorem 1 *The SPCA* (2) *admits an equivalent MISDP formulation*

$$\text{(SPCA)}\quad w^*:=\max_{\substack{\boldsymbol{z}\in Z,\\ \boldsymbol{X},\boldsymbol{W}_1,\cdots,\boldsymbol{W}_n\in\mathcal{S}_+^d}}\left\{\sum_{i\in[n]}\boldsymbol{c}_i^\top\boldsymbol{W}_i\boldsymbol{c}_i:\mathrm{tr}(\boldsymbol{X})=1,\boldsymbol{X}\succeq\boldsymbol{W}_i,\mathrm{tr}(\boldsymbol{W}_i)=z_i,\forall i\in[n]\right\}. \tag{6}$$

Proof. According to Part (ii) in Lemma 1, the largest eigenvalue of a symmetric matrix can be equivalently reformulated as an SDP, thus by introducing a positive semi-definite matrix variable $\boldsymbol{X}\in\mathcal{S}_+^d$, SPCA (5) can be represented as

$$w^*:=\max_{\boldsymbol{z}\in Z,\boldsymbol{X}\in\mathcal{S}_+^d}\left\{\sum_{i\in[n]}z_i\boldsymbol{c}_i^\top\boldsymbol{X}\boldsymbol{c}_i:\mathrm{tr}(\boldsymbol{X})=1\right\}, \tag{7}$$

where the objective function comes from the identity $\mathrm{tr}(\boldsymbol{c}_i\boldsymbol{c}_i^\top\boldsymbol{X})=\boldsymbol{c}_i^\top\boldsymbol{X}\boldsymbol{c}_i$ for each $i\in[n]$.

In SPCA (7), the objective function contains bilinear terms $\{z_i\boldsymbol{X}\}_{i\in[n]}$. To further convexify them, we create two copies of the matrix variable $\boldsymbol{X}$, denoting by $\boldsymbol{W}_{i1},\boldsymbol{W}_{i2}$ for each $i\in[n]$ and one of them will be equal to $\boldsymbol{X}$ depending on the value of binary variable z_i. Specifically, SPCA (7) now becomes

$$\begin{aligned}w^*:=\max_{\boldsymbol{z}\in Z,\boldsymbol{X},\boldsymbol{W}_{i1},\boldsymbol{W}_{i2}\in\mathcal{S}_+^d}\Bigg\{&\sum_{i\in[n]}\boldsymbol{c}_i^\top\boldsymbol{W}_{i1}\boldsymbol{c}_i:\boldsymbol{X}=\boldsymbol{W}_{i1}+\boldsymbol{W}_{i2},\forall i\in[n],\mathrm{tr}(\boldsymbol{X})=1,\\ &\mathrm{tr}(\boldsymbol{W}_{i1})=z_i,\mathrm{tr}(\boldsymbol{W}_{i2})=1-z_i,\forall i\in[n]\Bigg\}.\end{aligned}$$

Above, the matrix variables $\{\boldsymbol{W}_{i2}\}_{i\in[n]}$ are redundant and can be replaced by inequality $\boldsymbol{X}\succeq\boldsymbol{W}_i$ for each $i\in[n]$. Thus, we arrive at the equivalent reformulation (4) for SPCA. □

Theorem 1 presents the first equivalent MISDP formulation (6) to SPCA. The resulting formulation (6) has several interesting properties: (i) it can be directly solved via exact MISDP solvers such as YALMIP; (ii) matrix variables $\boldsymbol{X}$ and $\{\boldsymbol{W}_i\}_{i\in[n]}$ have dimension of $d\times d$, where d is the rank of matrix $\boldsymbol{A}$. Thus, the size of SPCA (6) can be further reduced if the covariance matrix $\boldsymbol{A}$ is low-rank; and (iii) the binary variables $\boldsymbol{z}$ can be separated from the other variables, so one can apply the Benders decomposition to solving the SPCA (6). This result will be elaborated with more details in the next subsection.

For large-scale instances, computing the continuous relaxation values of the SPCA (6) provides us an upper bound to the optimal value or can be useful to check the quality of different heuristics. In the following, we show that the continuous relaxation value of SPCA (6) is not too far away from the optimal value w^*. First, let $\overline{w}_1$ denote the continuous relaxation value, i.e.,

$$\overline{w}_1 := \max_{\substack{\boldsymbol{z}\in\overline{Z},\\ \boldsymbol{X},\boldsymbol{W}_1,\cdots,\boldsymbol{W}_n\in\mathcal{S}_+^d}} \left\{ \sum_{i\in[n]} \boldsymbol{c}_i^\top \boldsymbol{W}_i \boldsymbol{c}_i : \mathrm{tr}(\boldsymbol{X})=1, \boldsymbol{X}\succeq \boldsymbol{W}_i, \mathrm{tr}(\boldsymbol{W}_i)=z_i, \forall i\in[n] \right\}, \tag{8}$$

where we let $\overline{Z}$ denote the continuous relaxation of set Z, i.e.,

$$\overline{Z} = \left\{ \boldsymbol{z}\in[0,1]^n : \sum_{i\in[n]} z_i = k \right\}.$$

Theorem 2 *The continuous relaxation value $\overline{w}_1$ of formulation* (6) *achieves a* $\min\{k, n/k\}$ *optimality gap of SPCA, i.e.,*

$$w^* \le \overline{w}_1 \le \min\{k, n/k\} w^*.$$

Proof. It is obvious that $w^*\le\overline{w}_1$ since the feasible region of continuous relaxation (8) includes the original decision space. Thus, it remains to show that (i) $\overline{w}_1\le kw^*$ and (ii) $\overline{w}_1\le n/kw^*$.

Part (i) $\overline{w}_1\le kw^*$. For any feasible solution $(\boldsymbol{z},\boldsymbol{X},\{\boldsymbol{W}_i\}_{i\in[n]})$ to problem (8), we must have

$$\sum_{i\in[n]} \boldsymbol{c}_i^\top \boldsymbol{W}_i \boldsymbol{c}_i \le \sum_{i\in[n]} \boldsymbol{c}_i^\top \boldsymbol{c}_i \,\mathrm{tr}(\boldsymbol{W}_i) = \sum_{i\in[n]} z_i \boldsymbol{c}_i^\top \boldsymbol{c}_i \le \sum_{i\in[n]} z_i w^* = kw^*,$$

where the first inequality is due to the fact that the trace of the product of two symmetric positive semi-definite matrices is no larger than the product of the traces of these two matrices [10], the first equality is from $\mathrm{tr}(\boldsymbol{W}_i)=z_i$ for each $i\in[n]$, the second inequality is because

$$\boldsymbol{c}_i^\top \boldsymbol{c}_i = \lambda_{\max}\left(\boldsymbol{c}_i\boldsymbol{c}_i^\top\right) \le \max_{S\subseteq[n]:|S|=k} \lambda_{\max}\left(\sum_{j\in S} \boldsymbol{c}_j\boldsymbol{c}_j^\top\right) := w^*,$$

and the second equality is due to $\sum_{i\in[n]} z_i = k$.

Part (ii) $\overline{w}_1 \leq n/kw^*$. Similarly, given any feasible solution $(\boldsymbol{z}, \boldsymbol{X}, \{\boldsymbol{W}_i\}_{i\in[n]})$ of continuous relaxation (8), we must have

$$\sum_{i\in[n]} \boldsymbol{c}_i^\top \boldsymbol{W}_i \boldsymbol{c}_i \leq \sum_{i\in[n]} \boldsymbol{c}_i^\top \boldsymbol{X} \boldsymbol{c}_i = \frac{1}{\binom{n-1}{k-1}} \sum_{S\in\binom{[n]}{k}} \sum_{i\in S} \boldsymbol{c}_i^\top \boldsymbol{X} \boldsymbol{c}_i \leq \frac{\binom{n}{k}}{\binom{n-1}{k-1}} w^* = \frac{n}{k} w^*,$$

where the first inequality is because $\boldsymbol{W}_i \succeq \boldsymbol{X}$ and the second one is from Part (ii) in Lemma 1. □

Theorem 2 shows that the continuous relaxation value of formulation (8) is at most $\min\{k, n/k\}$ away from the true optimal value of SPCA (6), implying that if $k \to 1$ or $k \to n$, then the continuous relaxation value $\overline{w}_1$ is very close to the true optimal value w^*, which is consistent with the numerical study in Section 6.

2.2. Solving SPCA (6) and SDP Relaxation (8): Benders Decomposition It has been recognized that large-scale SDPs are challenging to solve, so is the MISDP (6). In this subsection, we apply the Benders decomposition [4, 18] to the proposed MISDP (6), which can be further integrated into the branch and cut framework. By relaxing the binary vector $\boldsymbol{z}$ to be continuous, the Benders Decomposition recasts the continuous SDP relaxation (8) as a maximin saddle point problem, which enables the adoption of the efficient subgradient method.

The main idea of Benders decomposition is to decompose SPCA (6) into two stages: first, the master problem is a pure integer maximization problem over $\boldsymbol{z}$, and second, given a feasible $\boldsymbol{z} \in Z$, the subproblem is to maximize over the remaining variables $(\boldsymbol{X}, \{\boldsymbol{W}_i\}_{i\in[n]})$. Thus, by separating the binary variables, we rewrite the SPCA (6) as

$$w^* := \max_{\boldsymbol{z}\in Z} H_1(\boldsymbol{z}) := \max_{\boldsymbol{X}, \boldsymbol{W}_1, \cdots, \boldsymbol{W}_d \in \mathcal{S}_+^d} \left\{ \sum_{i\in[n]} \boldsymbol{c}_i^\top \boldsymbol{W}_i \boldsymbol{c}_i : \mathrm{tr}(\boldsymbol{X}) = 1, \boldsymbol{X} \succeq \boldsymbol{W}_i, \mathrm{tr}(\boldsymbol{W}_i) = z_i, \forall i \in [n] \right\}. \quad (9)$$

Benders decomposition is of particular interest when the subproblem $H_1(\boldsymbol{z})$ for any $\boldsymbol{z} \in Z$ is easy to compute, which is, unfortunately, not the case. Therefore, it is desirable if we can specify the function $H_1(\boldsymbol{z})$ for any given $\boldsymbol{z} \in Z$ in an efficient way. Surprisingly, invoking Part(ii) in Lemma 1, the strong duality of inner SDP maximization problem in (9) holds and the obtained dual problem admits a closed-form solution for any binary variables $\boldsymbol{z} \in Z$, which enables the subproblem to generate valid inequalities to the master problem efficiently. The results are shown below.

Proposition 1 *For the function $H_1(\boldsymbol{z})$ defined in (9), we have*

(i) For any $\boldsymbol{z} \in \overline{Z}$, function $H_1(\boldsymbol{z})$ is equivalent to

$$H_1(\boldsymbol{z}) = \min_{\boldsymbol{\mu}, \boldsymbol{Q}_1, \cdots, \boldsymbol{Q}_n \in \mathcal{S}_+^d} \left\{ \lambda_{\max}\left(\sum_{i\in[n]} \boldsymbol{Q}_i \right) + \sum_{i\in[n]} \mu_i z_i : \boldsymbol{c}_i \boldsymbol{c}_i^\top \preceq \boldsymbol{Q}_i + \mu_i \boldsymbol{I}_d, 0 \leq \mu_i \leq \|\boldsymbol{c}_i\|_2^2, \forall i \in [n] \right\}, \quad (10)$$

which is concave in $\boldsymbol{z}$.

(ii) For any binary $\boldsymbol{z} \in Z$, an optimal solution to problem (10) is $\mu_i^* = 0$ if $z_i = 1$ and $\|\boldsymbol{c}_i\|_2^2$, otherwise, and $\boldsymbol{Q}_i^* := (1 - \mu_i^*/\|\boldsymbol{c}_i\|_2^2)\boldsymbol{c}_i\boldsymbol{c}_i^\top$ for each $i \in [n]$.

Proof. See Appendix A.2. □

The Part (ii) of Proposition 1 shows that given a solution $\boldsymbol{z} \in Z$ with its support S, the optimal value to (10) is equal to

$$H_1(\boldsymbol{z}) = \lambda_{\max}\bigg(\sum_{i\in S} \boldsymbol{c}_i\boldsymbol{c}_i^\top\bigg) + \sum_{i\in[n]\setminus S} ||\boldsymbol{c}_i||_2^2,$$

which leads to an equivalent reformulation of SPCA (9) as

$$w^* = \max_{\boldsymbol{z}\in Z}\bigg\{w : w \le \lambda_{\max}(\boldsymbol{A}_{SS}) + \sum_{i\in[n]\setminus S} \|\boldsymbol{c}_i\|_2^2 z_i, \forall S \subseteq [n] : |S| = k\bigg\}. \tag{11}$$

Above, for any mixed binary solution $(\widehat{\boldsymbol{z}}, \widehat{w}) \in Z \times \mathbb{R}$, the most violated constraint is

$$w \le \lambda_{\max}(\boldsymbol{A}_{\widehat{S}\widehat{S}}) + \sum_{i\in[n]\setminus \widehat{S}} \|\boldsymbol{c}_i\|_2^2 z_i,$$

where set $\widehat{S} := \{i \in [n] : \widehat{z}_i = 1\}$ denotes the support of $\widehat{\boldsymbol{z}}$. We remark that the exact branch and cut approach to solve SPCA (11) using *callback* functions will benefit from these closed-form valid inequalities.

Note that by relaxing the binary variables to be continuous, the relaxed problem (9) is equivalent to the SDP relaxation (8). However, given $z \in \overline{Z}$, the dual representation of function $H_1(\boldsymbol{z})$ in (10) is still a difficult SDP. Motivated by Part (ii) in Proposition 1, we propose a more efficient upper bound $\overline{H}_1(\boldsymbol{z})$ than $H_1(\boldsymbol{z})$ by letting $\boldsymbol{Q}_i := (1 - \mu_i/\|\boldsymbol{c}_i\|_2^2)\boldsymbol{c}_i\boldsymbol{c}_i^\top$ for each $i \in [n]$ to problem (10). In the next theorem, we show that the relaxed $\overline{H}_1(\boldsymbol{z})$ becomes exact for any binary vector $\boldsymbol{z} \in Z$ and the resulting upper bound of SPCA also achieves a $\min\{k, n/k\}$ optimality gap.

Theorem 3 *The following results hold for the relaxed function $\overline{H}_1(\boldsymbol{z})$:*

(i) For any $\boldsymbol{z} \in \overline{Z}$, function $H_1(\boldsymbol{z})$ is upper bounded by

$$\overline{H}_1(\boldsymbol{z}) = \min_{\boldsymbol{\mu}}\bigg\{\lambda_{\max}\bigg(\sum_{i\in[n]}(1 - \mu_i/\|\boldsymbol{c}_i\|_2^2)\boldsymbol{c}_i\boldsymbol{c}_i^\top\bigg) + \sum_{i\in[n]}\mu_i z_i : 0 \le \mu_i \le \|\boldsymbol{c}_i\|_2^2, \forall i \in [n]\bigg\}; \tag{12}$$

(ii) If $\boldsymbol{z} \in Z$, then $H_1(\boldsymbol{z}) = \overline{H}_1(\boldsymbol{z}) = \lambda_{\max}(\sum_{i\in[n]} z_i\boldsymbol{c}_i\boldsymbol{c}_i^\top)$; and

(iii) The continuous relaxation value of SPCA

$$\overline{w}_2 = \max_{\boldsymbol{z}\in\overline{Z}} \overline{H}_1(\boldsymbol{z}) \tag{13}$$

achieves a $\min\{k, n/k\}$ optimality gap of SPCA, i.e., $w^* \le \overline{w}_1 \le \overline{w}_2 \le \min\{k, n/k\}w^*$, where $\overline{w}_1$ is defined in (8).

Proof.

(i) The conclusion follows by choosing a feasible $\boldsymbol{Q}_i := (1-\mu_i/\|\boldsymbol{c}_i\|_2^2)\boldsymbol{c}_i\boldsymbol{c}_i^\top$ for each $i\in[n]$ in the representation (10).

(ii) For any $\boldsymbol{z}\in Z$, we derive from Part (ii) in Proposition 1 that $\overline{H}_1(\boldsymbol{z}) \geq \lambda_{\max}(\sum_{i\in[n]} z_i\boldsymbol{c}_i\boldsymbol{c}_i^\top)$. Thus, it is sufficient to show that $\overline{H}_1(\boldsymbol{z}) \leq \lambda_{\max}(\sum_{i\in[n]} z_i\boldsymbol{c}_i\boldsymbol{c}_i^\top)$. Indeed, this can be done simply by letting $\mu_i = 0$ if $z_i = 0$, and $||\boldsymbol{c}_i||_2^2$, otherwise in (12).

(iii) By the proof of Theorem 2, to obtain the same optimality gap for (13) as SDP (8), we need to show that $\overline{H}_1(\boldsymbol{z}) \leq \sum_{i\in[n]} z_i\boldsymbol{c}_i^\top\boldsymbol{c}_i$ and $\overline{H}_1(\boldsymbol{z}) \leq \lambda_{\max}(\boldsymbol{A}) = \lambda_{\max}(\sum_{i\in[n]}\boldsymbol{c}_i\boldsymbol{c}_i^\top)$ for any $\boldsymbol{z}\in\overline{Z}$.
We must have $\overline{H}_1(\boldsymbol{z}) \leq \sum_{i\in[n]} z_i\boldsymbol{c}_i^\top\boldsymbol{c}_i$ by by letting $\mu_i = \boldsymbol{c}_i^\top\boldsymbol{c}_i$ for all $i\in[n]$ in (12).
We also have $\overline{H}_1(\boldsymbol{z}) \leq \lambda_{\max}(\boldsymbol{A}) = \lambda_{\max}(\sum_{i\in[n]}\boldsymbol{c}_i\boldsymbol{c}_i^\top)$ by letting $\mu_i = 0$ for all $i\in[n]$ in (12).
Then the rest of the proof follows directly from that of Theorem 2 and is thus omitted.

□

We remark that: (i) Compared to $H_1(\boldsymbol{z})$, function $\overline{H}_1(\boldsymbol{z})$ in (12) only involves an n-dimensional variable $\boldsymbol{\mu}$. The resulting relaxation (13) of SPCA can be viewed as a conventional saddle problem so we apply the subgradient method with convergence rate of $O(1/T)$ to the search for optimal solutions (see, e.g., [31]), which offers an efficient way to generate an upper bound of SPCA in Section 6; (ii) On the other hand, the continuous relaxation value $\overline{w}_1 = \max_{\boldsymbol{z}\in\overline{Z}} H_1(\boldsymbol{z})$ tends to be stronger than $\overline{w}_2$ in (13). Thus, it is a tradeoff between computational effort and a better upper bound; (iii) Surprisingly, both bounds $\overline{w}_1, \overline{w}_2$ achieve the same optimality gap of SPCA. This implies that there might be room to improve the analysis of optimality gap in Theorem 2. We leave this to interested readers; and (iv) more importantly, when $\boldsymbol{z}\in Z$ is binary, both problems (10) and (12) have closed-form results, which are very helpful for using the branch and cut method.

3. Exact MISDP Formulation (II) The MISDP formulation (6) developed for SPCA in the previous section mainly are inspired from Part(ii) and Part(iii) in Lemma 1. In this section, we will propose another exact MISDP reformulation of SPCA using Part(i) and Part(ii) in Lemma 1. Similarly, we will present the optimality gap of the corresponding SDP relaxation to demonstrate the strength of the second formulation. It is worthy of noting that the proposed MISDP (6) requires the positive semi-definiteness of matrix $\boldsymbol{A}$ as it is built on Cholesky decomposition of $\boldsymbol{A}$, but the result in this section is more general and holds even matrix $\boldsymbol{A}$ is not positive semi-definite.

3.1. A Naive Exact MISDP Formulation We first establish a naive exact MISDP formulation of SPCA (2) based on Part (ii) in Lemma 1, and the resulting continuous relaxation value is equal to $\lambda_{\max}(\boldsymbol{A})$.

Proposition 2 *The SPCA* (2) *admits the following MISDP formulation:*

$$\text{(SPCA)} \quad w^* := \max_{\boldsymbol{z}\in Z, \boldsymbol{X}\in\mathcal{S}^n_+} \left\{ \text{tr}(\boldsymbol{A}\boldsymbol{X}) : \text{tr}(\boldsymbol{X}) = 1, X_{ii} \le z_i, \forall i \in [n] \right\}. \tag{14}$$

and its continuous relaxation value is equal to $\lambda_{\max}(\boldsymbol{A})$.

Proof. See Appendix A.3. □

The SPCA formulation (14) can be also found in [17]. However, our proof is quite different and shorter, since it does not involve sophisticated extreme point characterization of SDPs. Although the MISDP (14) is equivalent to SPCA (2), the fact that its continuous relaxation value is equal to $\lambda_{\max}(\boldsymbol{A})$ demonstrates that it might be a weak formulation. This motivates us to further strengthen the formulation (14) by adding valid inequalities in the next subsection.

3.2. A Stronger Reformulation with Two Valid Inequalities In this subsection, we first propose two valid inequalities for SPCA (14) and derive the optimality gap of its continuous relaxation value of the improved formulation.

After examining different types of valid inequalities, we propose the following two types of valid inequalities for the SPCA formulation (14).

Lemma 2 *The following two inequalities are valid to SPCA* (14)

(i) $\sum_{j\in[n]} X_{ij}^2 \le X_{ii} z_i$ *for all* $i \in [n]$*; and*

(ii) $\left(\sum_{j\in[n]} |X_{ij}|\right)^2 \le k X_{ii} z_i$ *for all* $i \in [n]$.

Proof. See Appendix A.4. □

We make the following remarks about Lemma 2.

(i) Many other valid inequalities are dominated by the two types of valid inequalities in Lemma 2 such as

$$|X_{ij}| \le z_i, X_{ij}^2 \le X_{ii} z_j, X_{ij}^2 \le z_i z_j, \forall i, j \in [n];$$

(ii) Note that the two types of valid inequalities are both second order conic (see e.g., [3]), and thus can be embedded into SDP solvers such as MOSEK, SDPT3; and

(iii) We further observe that the inequality $X_{ii} \le z_i$ in (14) is dominated by the first type of inequalities with the facts that $X_{ii}^2 + \sum_{j\in[n]\setminus\{i\}} X_{ij}^2 \le X_{ii} z_i$ and $X_{ii} \ge 0$ for each $i \in [n]$.

The results in Lemma 2 together with Proposition 2 give rise to a stronger MISDP of SPCA than formulation (14), which is summarized below.

Theorem 4 *The SPCA* (2) *can reduce to following stronger MISDP formulation:*

$$\text{(SPCA)}\ w^* := \max_{\boldsymbol{z}\in Z, \boldsymbol{X}\in\mathcal{S}^n_+} \left\{ \text{tr}(\boldsymbol{A}\boldsymbol{X}) : \text{tr}(\boldsymbol{X}) = 1, \sum_{j\in[n]} X_{ij}^2 \le X_{ii} z_i, \left(\sum_{j\in[n]} |X_{ij}|\right)^2 \le k X_{ii} z_i, \forall i \in [n] \right\}. \tag{15}$$

Let $\overline{w}_3$ denote the continuous relaxation value of SPCA formulation (15), i.e.,

$$\overline{w}_3 := \max_{\boldsymbol{z}\in\overline{Z},\boldsymbol{X}\in\mathcal{S}_+^n}\left\{\operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X})=1, \sum_{j\in[n]} X_{ij}^2 \le X_{ii}z_i, \left(\sum_{j\in[n]}|X_{ij}|\right)^2 \le kX_{ii}z_i, \forall i\in[n]\right\}. \tag{16}$$

Clearly, we have $\lambda_{\max}(\boldsymbol{A}) \ge \overline{w}_3$. We are going to prove that the continuous relaxation value can be even stronger than a well-known SDP upper bound for SPCA (2) introduced by d'Aspremont et al. [13], denoted by $\overline{w}_4$, that has been widely used for solving SPCA in literature. The upper bound from [13] comes to the following formulation

$$\overline{w}_4 := \max_{\boldsymbol{X}\in\mathcal{S}_+^n}\left\{\operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X})=1, \sum_{i\in[n]}\sum_{j\in[n]}|X_{ij}| \le k\right\}. \tag{17}$$

The formal comparison result is shown below.

Proposition 3 *The upper bounds $\overline{w}_3, \overline{w}_4$ of SPCA defined in* (16) *and*(17)*, respectively, satisfy $\overline{w}_4 \ge \overline{w}_3$, i.e., the continuous relaxations value of the stronger MISDP* (15) *is stronger than the optimal value of the SDP formulation* (17) *from [13].*

Proof. To show that $\overline{w}_4 \ge \overline{w}_3$, it is sufficient to prove that any feasible solution $(\boldsymbol{z}, \boldsymbol{X})$ of the continuous relaxation problem (16), will satisfy the constraints in the SDP formulation (17).

Clearly, we have $\boldsymbol{X} \in \mathcal{S}_+^n$ and $\operatorname{tr}(\boldsymbol{X}) = 1$. It remains that $\sum_{i\in[n]}\sum_{j\in[n]}|X_{ij}| \le k$. Indeed, we have

$$\sum_{i\in[n]}\sum_{j\in[n]}|X_{ij}| \le \sum_{i\in[n]}\sqrt{k}\sqrt{X_{ii}z_i} \le \sqrt{k}\sqrt{\sum_{i\in[n]}X_{ii}}\sqrt{\sum_{i\in[n]}z_i} = k,$$

where the first inequality results from type (ii) inequalities in Lemma 2, the second one is due to Cauchy–Schwartz inequality, and the equality is due to $\operatorname{tr}(\boldsymbol{X}) = 1$ and $\sum_{i\in[n]} z_i = k$. □

Next, we show that the continuous relaxations value of the stronger MISDP (15) is also quite close to the true value. This phenomenon is more striking in the numerical study.

Theorem 5 *The continuous relaxations value of the stronger MISDP formulation* (15) *yields a* $\min\{k, n/k\}$ *optimality gap for SPCA, i..e,*

$$w^* \le \overline{w}_3 \le \min\{k, n/k\}w^*.$$

Proof. The proof is separated into two parts: (i) $\overline{w}_3 \le kw^*$ and (ii) $\overline{w}_3 \le n/kw^*$.

(i) $\overline{w}_3 \le kw^*$. For any feasible solution $\boldsymbol{X}$ to problem (16), we have

$$\operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) = \sum_{i\in[n]}\sum_{j\in[n]}A_{ij}X_{ij} \le \sum_{i\in[n]}\sum_{j\in[n]}|A_{ij}||X_{ij}| \le w^*\sum_{i\in[n]}\sum_{j\in[n]}|X_{ij}| \le kw^*,$$

where the first inequality is due to taking the absolute values, the second one is based on the fact that $\max_{i\in[n]}\{A_{i,i}\} \le w^*$ and $|A_{i,j}| \le \sqrt{A_{i,i}A_{j,j}} \le w^*$ for each pair $i, j \in [n]$, and the third one can be obtained from the proof of Proposition 3.

(ii) $\overline{w}_3 \leq n/kw^*$. The proof is similar to the one of Theorem 2 since $\overline{w}_3 \leq \lambda_{\max}(\boldsymbol{A}) \leq n/kw^*$. $\square$

In general, our two proposed MISDP formulations (6) and (15) are not comparable although their continuous relaxations have the same theoretical approximation gap, which will be also illustrated in the numerical study section. The continuous relaxation of the MISDP formulation (15) might be difficult to solve due to lager size of its matrix variables and higher complexity of its constraints. In the next subsection, we will discuss Benders decomposition for SPCA (15), where the subproblem reduces to a second order conic program rather than an SDP.

3.3. Benders Decomposition The decomposition method developed for SPCA (15) in this subsection follows from Section 2.2. Therefore, many details will be omitted for brevity. Similarly, we decompose the proposed MISDP formulation (15) by a master problem over binary variables $\boldsymbol{z} \in Z$ and a subproblem over the matrix variable $\boldsymbol{X} \in \mathcal{S}_+^n$. Also, we reformulate SPCA (15) as the following equivalent two-stage optimization problem

$$w^* = \max_{\boldsymbol{z}\in Z} H_2(\boldsymbol{z}) := \max_{\boldsymbol{X}\in\mathcal{S}_+^n} \left\{ \operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X}) = 1, \sum_{j\in[n]} X_{ij}^2 \leq X_{ii} z_i, \left(\sum_{j\in[n]} |X_{ij}|\right)^2 \leq k X_{ii} z_i, \forall i \in [n] \right\}. \tag{18}$$

It is favorable to derive an efficient dual formulation of $H_2(\boldsymbol{z})$ for any given $\boldsymbol{z} \in \overline{Z}$ such that its subgradient can be easily computed. Indeed, invoking Part(ii) in Lemma 1 and dualizing the second order conic constraints, the strong duality of inner maximization over $\boldsymbol{X}$ in (18) still holds. The proof is similar to Proposition 1 and is thus omitted.

Proposition 4 *For any $\boldsymbol{z} \in \overline{Z}$, function $H_2(\boldsymbol{z})$ is equivalent to*

$$\begin{aligned} H_2(\boldsymbol{z}) = \min_{\boldsymbol{\mu},\boldsymbol{\nu}_1,\boldsymbol{\nu}_2,\boldsymbol{\Lambda},\boldsymbol{W}_1,\boldsymbol{W}_2,\boldsymbol{\beta}} & \lambda_{\max}\left(\boldsymbol{A} + \boldsymbol{\Lambda} + 1/2\operatorname{Diag}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 + \boldsymbol{\nu}_1 + \boldsymbol{\nu}_2) - \boldsymbol{W}_1 + \boldsymbol{W}_2\right) \\ & + 1/2(-\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)^\top \boldsymbol{z} + k/2(-\boldsymbol{\nu}_1 + \boldsymbol{\nu}_2)^\top \boldsymbol{z}, \\ \text{s.t.} \quad & \beta_i + (\boldsymbol{W}_1)_{ij} + (\boldsymbol{W}_2)_{ij} \leq 0, \forall i \in [n], j \in [n], \\ & \sum_{j\in[n]} \boldsymbol{\Lambda}_{ij}^2 + (\mu_{i1})^2 \leq (\mu_{i2})^2, \forall i \in [n], \\ & \beta_i^2 + (\nu_{i1})^2 \leq (\nu_{i2})^2, \forall i \in [n], \\ & (W_1)_{ij} \geq 0, (W_2)_{ij} \geq 0, \forall i \in [n], \forall j \in [n], \\ & \boldsymbol{\nu}_1, \boldsymbol{\nu}_2 \in \mathbb{R}_+^n, \boldsymbol{\Lambda}, \boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathcal{S}^n, \end{aligned} \tag{19}$$

which is concave in $\boldsymbol{z}$.

For the equivalent function $H_2(\boldsymbol{z})$ derived in Proposition 4, we remark that: (i) Note that for any given $\boldsymbol{z} \in \overline{Z}$, function $H_2(\boldsymbol{z})$ can be solved as an second order conic program and escape from the SDP curse. More effectively, it can be solved via many first-order methods (e.g., the subgradient

method) since the subgradient is easy to obtain and the projection only involves second order conic constraints; (ii) On the other hand, when we solve the continuous relaxation

$$\overline{w}_3 = \max_{\boldsymbol{z}\in\overline{Z}} H_2(\boldsymbol{z}), \tag{20}$$

the subgradient method is also applicable to solve the entire maximin saddle problem with $O(1/T)$ rate of convergence (see, e.g., [31]); (iii) We can warm start the exact branch and cut algorithm by solving the continuous relaxation (20), and add all the subgradient inequalities into the root relaxed problem.

4. A Mixed-Integer Linear Program (MILP) for SPCA with Arbitrary Accuracy

The formulations developed in the previous section for solving SPCA either rely on MISDP solvers or customized branch and cut algorithms, which does not leverage existing computational powers of solvers such as CPLEX, Gurobi. In this section, motivated by the SPCA formulation (5) and the identity of eigenvalues, we further derive an approximate mixed-integer linear program (MILP) for SPCA with arbitrary accuracy $\epsilon > 0$ and $O(n + d + \log(\epsilon^{-1}))$ binary variables. We also prove the optimality gap of its corresponding LP relaxation. The results in this section assume that $\boldsymbol{A}$ is positive semi-definite.

4.1. An MILP Formulation for SPCA

The difficulty of SPCA (5) lies in how to convexify the objective function, i.e., the largest eigenvalue of a symmetric matrix $\boldsymbol{A}$. In particular, our proposed MISDP formulations stem from the fact that the largest eigenvalue can be formulated as an equivalent SDP problem. Through a different lens, we represent the largest eigenvalue function based on the natural definition of eigenvalues of a matrix, i.e.,

$$\lambda_{\max}(\boldsymbol{A}) = \max_{w,\boldsymbol{x}\in\mathbb{R}^n} \Big\{ w : \boldsymbol{A}\boldsymbol{x} = w\boldsymbol{x}, \boldsymbol{x} \neq 0 \Big\},$$

where $\boldsymbol{x}$ denotes an eigenvector and the nonzero constraint rules out the trivial solution $\boldsymbol{x} = 0$.

This motivates us to recast SPCA formulation (5) as the following nonconvex problem

$$w^* = \max_{w,\boldsymbol{x}\in\mathbb{R}^d,\boldsymbol{z}\in Z} \Big\{ w : \sum_{i\in[n]} z_i \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{x} = w\boldsymbol{x}, \|\boldsymbol{x}\|_\infty = 1 \Big\}, \tag{21}$$

where $\|\boldsymbol{x}\|_\infty = 1$ also excludes the trivial solution $\boldsymbol{x} = \boldsymbol{0}$.

For any given $\boldsymbol{z} \in Z$, the nonconvexity of SPCA formulation (21) lies in three aspects: (i) Bilinear terms $\{z_i\boldsymbol{x}\}_{i\in[n]}$. They can be easily linearized using the disjunctive programming techniques since vector $\boldsymbol{z}$ is binary; (ii) Constraint $\|\boldsymbol{x}\|_\infty = 1$. The nonconvex constraint $\|\boldsymbol{x}\|_\infty = 1$ can be equivalently written as a disjunction with $2d$ sets below

$$\cup_{j\in[d]} \big\{ \boldsymbol{x} \in \mathbb{R}^d : x_j = 1, \|\boldsymbol{x}\|_\infty \leq 1 \big\} \cup_{j\in[d]} \big\{ \boldsymbol{x} \in \mathbb{R}^d : x_j = -1, \|\boldsymbol{x}\|_\infty \leq 1 \big\}.$$

Due to the equivalence of $\boldsymbol{x}$ and $-\boldsymbol{x}$ in SPCA (21), it suffices to only keep first d sets, i.e., $\cup_{j\in[d]}\{\boldsymbol{x}\in\mathbb{R}^d : x_j=1, \|\boldsymbol{x}\|_\infty\le 1\}$. This disjunction can be equivalently described as an MILP using the results in [2]; and (iii) Bilinear term $w\boldsymbol{x}$. We can first approximate variable w using binary expansion and then linearize the obtained bilinear terms by the same disjunctive technique as part (i). The resulting MILP formulation is summarized in the following theorem.

Theorem 6 *Given a threshold $\epsilon>0$, the following MILP is $O(\epsilon)$-approximate to SPCA* (2)*, i.e., $\epsilon\le\widehat{w}(\epsilon)-w^*\le\epsilon\sqrt{d}$*

$$
\begin{aligned}
\widehat{w}(\epsilon):=\max_{w,\boldsymbol{z}\in Z,\boldsymbol{y},\boldsymbol{\alpha},\boldsymbol{x},,\boldsymbol{\delta},\boldsymbol{\mu},\boldsymbol{\sigma}} & \ w\\
\text{s.t.}\quad & \boldsymbol{x}=\boldsymbol{\delta}_{i1}+\boldsymbol{\delta}_{i2}, ||\boldsymbol{\delta}_{i1}||_\infty\le z_i, ||\boldsymbol{\delta}_{i2}||_\infty\le 1-z_i, \forall i\in[n],\\
& \boldsymbol{x}=\sum_{j\in[d]}\boldsymbol{\sigma}_j, ||\boldsymbol{\sigma}_j||_\infty\le y_j, \sigma_{jj}=y_j, \forall j\in[d], \sum_{j\in[d]}y_j=1,\\
& \boldsymbol{x}=\boldsymbol{\mu}_{\ell 1}+\boldsymbol{\mu}_{\ell 2}, ||\boldsymbol{\mu}_{\ell 1}||_\infty\le\alpha_\ell, ||\boldsymbol{\mu}_{\ell 2}||_\infty\le 1-\alpha_\ell, \forall\ell\in[m],\\
& w=w_U-(w_U-w_L)\Big(\sum_{i\in[m]}2^{-i}\alpha_i\Big),\\
& \Big|\Big|\sum_{i\in[n]}\boldsymbol{c}_i\boldsymbol{c}_i^\top\boldsymbol{\delta}_{i1}-w_U\boldsymbol{x}+(w_U-w_L)\sum_{\ell\in[m]}2^{-\ell}\boldsymbol{\mu}_{\ell 1}\Big|\Big|_\infty\le\epsilon,\\
& \boldsymbol{\alpha}\in\{0,1\}^m, \boldsymbol{y}\in\{0,1\}^d,
\end{aligned}
\tag{22}
$$

where w_L, w_U separately denote the lower and upper bounds of SPCA, $m:=\lceil\log_2((w_U-w_L)\epsilon^{-1})\rceil$ and the infinite norm inequality constraints can be easily linearized.

Proof. See Appendix A.5. □

For the proposed MILP formulation (22), we remark that

(i) This is the first-known MILP representation with arbitrary accuracy $O(\epsilon)$ in literature of SPCA;

(ii) The MILP formulation (22), although compact, involves $O(n+d+\log\epsilon^{-1})$ binary variables, $O(nd+d\log\epsilon^{-1})$ continuous variables, and $O(nd+n\log\epsilon^{-1})$ linear constraints;

(iii) In SPCA (21), one might be curious about the choice of infinite norm. Unfortunately, as far as we are concerned, this is the only norm that leads to a compact MILP formulation;

(iv) In the MILP formulation (22), one might consider replacing the infinite norm in the constraint $||\sum_{i\in[n]}\boldsymbol{c}_i\boldsymbol{c}_i^\top\boldsymbol{\delta}_{i1}-w_U\boldsymbol{x}+(w_U-w_L)\sum_{i\in[m]}2^{-i}\boldsymbol{\mu}_{i1}||_\infty\le\epsilon$ by other norms, which will lead to different formulations (either MILP or mixed-integer conic program) and slightly different approximation bounds;

(v) Strong lower and upper bounds of SPCA w_L, w_U can speed up the solution procedure; and

(vi) Instead of building a relatively large-scale MILP formulation (22), one might solve d number of smaller-scale MILPs by enumerating each set of a disjunction $\cup_{j\in[d]}\{\boldsymbol{x}: x_j=1, \|\boldsymbol{x}\|_\infty\le 1\}$.

The last remark is summarized in the following corollary.

Corollary 1 *Given a threshold $\epsilon > 0$, the optimal value of MILP* (22) *is equal to $\widehat{w}(\epsilon) = \max_{j\in[d]} \widehat{w}_j(\epsilon)$, where for each $j \in [d]$, $\widehat{w}_j(\epsilon)$ is defined as*

$$\begin{aligned}
\widehat{w}_j(\epsilon) := \max_{w,\boldsymbol{z}\in Z,\boldsymbol{y},\boldsymbol{\alpha},\boldsymbol{x},\boldsymbol{\delta},\boldsymbol{\mu}} \; & w \\
\text{s.t.} \quad & \boldsymbol{x} = \boldsymbol{\delta}_{i1} + \boldsymbol{\delta}_{i2}, ||\boldsymbol{\delta}_{i1}||_\infty \le z_i, ||\boldsymbol{\delta}_{i2}||_\infty \le 1 - z_i, \forall i \in [n], \\
& ||\boldsymbol{x}||_\infty \le 1, x_j = 1, \\
& \boldsymbol{x} = \boldsymbol{\mu}_{\ell 1} + \boldsymbol{\mu}_{\ell 2}, ||\boldsymbol{\mu}_{\ell 1}||_\infty \le \alpha_\ell, ||\boldsymbol{\mu}_{\ell 2}||_\infty \le 1 - \alpha_\ell, \forall \ell \in [m], \\
& w = w_U - (w_U - w_L)\Big(\sum_{i\in[m]} 2^{-i}\alpha_i\Big), \\
& \Big|\Big| \sum_{i\in[n]} \boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1} - w_U\boldsymbol{x} + (w_U - w_L)\sum_{i\in[m]} 2^{-i}\boldsymbol{\mu}_{i1} \Big|\Big|_\infty \le \epsilon, \\
& \boldsymbol{\alpha} \in \{0,1\}^m,
\end{aligned} \tag{23}$$

where w_L, w_U separately denote the lower and upper bounds of SPCA, $m := \lceil \log_2((w_U - w_L)\epsilon^{-1})\rceil$ and the infinite norm inequality constraints can be easily linearized.

Albeit being smaller-size, some MILPs defined in Corollary 1 might be infeasible. Since the optimal value of an infeasible maximization problem is $-\infty$ by default, the result in Corollary 1 still holds. However, one might need to be cautious when using this result and be aware of infeasibilities.

4.2. Theoretical Optimality Gap Similar to other two exact formulations, we are also interested in deriving theoretical approximation bound for MILP formulation (22) by relaxing binary variables $\boldsymbol{z}$. Particularly, we assume that other binary variables $\boldsymbol{y}, \boldsymbol{\alpha}$ can be enumerated effectively. Our results show that the theoretical optimality gap is, in general, worse than the other two bounds.

Theorem 7 *Given a threshold $\epsilon > 0$, by enforcing the binary variables $\boldsymbol{z}$ to be continuous, let $\overline{w}_5(\epsilon)$ denote the optimal value of the relaxed MILP formulation* (22)*. Then we have*

$$\overline{w}_5(\epsilon) \le \min\big\{k(\sqrt{d}/2 + 1/2),\ n/k\sqrt{d} + (n-k)(\sqrt{d}/2 + 1/2)\big\}w^* + \epsilon\sqrt{d}.$$

Proof. See Appendix A.6. □

5. Approximation Algorithms In this section, motivated by the equivalent combinatorial formulation (4), we prove and demonstrate the tightness of the approximation ratios of the well-known greedy and local search algorithms for solving SPCA.

5.1. Greedy Algorithm The greedy algorithm has been widely used in many combinatorial problems with the cardinality constraint. The greedy algorithm in this subsection is particularly based on the combinatorial formulation (4), which proceeds as follows: Given a subset $\widehat{S}_G \subseteq [n]$ denoting the selected vectors, it aims to find a new vector from $\{\boldsymbol{c}_i\}_{i\in[n]\setminus\widehat{S}_G}$ to maximize the largest eigenvalue of the sum of rank-one matrices obtained so far including the new one. The detailed implementation can be found in Algorithm 1.

Algorithm 1 Greedy Algorithm for SPCA (4)

1: **Input:** $n\times n$ matrix $\boldsymbol{A}\succeq 0$ of rank d and integer $k\in[n]$
2: Let $\boldsymbol{A}=\boldsymbol{C}^\top\boldsymbol{C}$ denote its Cholesky factorization where $\boldsymbol{C}\in\mathbb{R}^{d\times n}$
3: Let $\boldsymbol{c}_i\in\mathbb{R}^d$ denote the i-th column vector of matrix $\boldsymbol{C}$ for each $i\in[n]$
4: Let $\widehat{S}_G:=\emptyset$ denote the chosen set
5: **for** $\ell=1,\cdots,k$ **do**
6: Compute $j^*\in\arg\max_{j\in[n]\setminus\widehat{S}}\{\lambda_{\max}(\sum_{i\in\widehat{S}_G\cup\{j\}}\boldsymbol{c}_i\boldsymbol{c}_i^\top)\}$
7: Add j^* to the set $\widehat{S}_G$
8: **end for**
9: **Output:** $\widehat{S}_G$

The following result show that the greedy Algorithm 1 yields $1/k$-approximation ratio.

Theorem 8 *The greedy Algorithm 1 yields a k^{-1}-approximation ratio for SPCA (4), i.e., the output $\widehat{S}_G$ of Algorithm 1 satisfies*

$$\lambda_{\max}\bigg(\sum_{i\in\widehat{S}_G}\boldsymbol{c}_i\boldsymbol{c}_i^\top\bigg)\geq\frac{1}{k}w^*.$$

Proof. Suppose that the optimal set of SPCA (4) is S^*, then we have

$$\lambda_{\max}\bigg(\sum_{i\in S^*}\boldsymbol{c}_i\boldsymbol{c}_i^\top\bigg)\leq\sum_{i\in S^*}\lambda_{\max}(\boldsymbol{c}_i\boldsymbol{c}_i^\top)\leq k\max_{i\in[n]}\lambda_{\max}(\boldsymbol{c}_i\boldsymbol{c}_i^\top)\leq k\lambda_{\max}\bigg(\sum_{i\in\widehat{S}_G}\boldsymbol{c}_i\boldsymbol{c}_i^\top\bigg),$$

where the first inequality results from the convexity of largest eigenvalue function and the last one is because at the first iteration, the greedy Algorithm 1 must choose the largest-length vector. □
The approximation ratio k^{-1} of greedy Algorithm 1 is tight, since there exists an example whose greedy optimum is no better than k^{-1}. This example is presented as below.

Example 1 *For any integer $k\in[d]$, let $d=k+1$, $n=2k$, and the vectors $\{\boldsymbol{c}_i\}_{i\in[n]}\subseteq\mathbb{R}^d$ be*

$$\boldsymbol{c}_i=\begin{cases}\boldsymbol{e}_i, & \text{if } i\in[k],\\ \boldsymbol{e}_{k+1}, & \text{if } i\in[k+1,n],\end{cases}\forall i\in[n].$$

Proposition 5 *In Example 1, the output value of greedy Algorithm 1 is k^{-1}-away from the true optimal value of SPCA. That is, approximation ratio k^{-1} of greedy Algorithm 1 is tight.*

Proof. In Example 1, according to the greedy Algorithm 1, it will select $\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_k$ at each iteration, i.e., the output set is $\widehat{S}_G = [k]$. Thus, the resulting largest eigenvalue of greedy Algorithm 1 is equal to 1.

Apparently, the true optimal value of Example 1 is equal to

$$\lambda_{\max}\bigg(\sum_{i\in[k+1,n]} \boldsymbol{c}_i\boldsymbol{c}_i^\top\bigg) = \lambda_{\max}\left(k\boldsymbol{e}_{k+1}\boldsymbol{e}_{k+1}^\top\right) = k.$$

This completes the proof. □

5.2. Local Search Algorithm The local search algorithm can improve the existing solutions and has been successfully used to solve many interesting machine learning and data analytics problems, such as experimental design [26] and maximum entropy sampling [24]. This subsection investigates the local search algorithm for SPCA (4) and proves its approximation ratio.

In the local search algorithm, we start with a size-k subset, and in each iteration, swap an element of chosen set with one of the unchosen set as long as it improves the largest eigenvalue. The detailed implementation can be found in Algorithm 2.

Algorithm 2 Local Search Algorithm for SPCA (4)

Input: $n \times n$ matrix $\boldsymbol{A} \succeq 0$ of rank d and integer $k \in [n]$
Let $\boldsymbol{A} = \boldsymbol{C}^\top\boldsymbol{C}$ denote its Cholesky factorization where $\boldsymbol{C} \in \mathbb{R}^{d\times n}$
Let $\boldsymbol{c}_i \in \mathbb{R}^d$ denote the i-th column vector of matrix $\boldsymbol{C}$ for each $i \in [n]$
Initialize a size-k subset $\widehat{S}_L \subseteq [n]$
do
    for each pair $(i,j) \in \widehat{S}_L \times ([n] \setminus \widehat{S}_L)$ **do**
        if $\lambda_{\max}\left(\sum_{\ell\in\widehat{S}_L\cup\{j\}\setminus\{i\}} \boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\right) > \lambda_{\max}\left(\sum_{\ell\in\widehat{S}_L} \boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\right)$ **then**
            Update $\widehat{S}_L := \widehat{S}_L \cup \{j\} \setminus \{i\}$
        end if
    end for
while there is still an improvement
Output: $\widehat{S}_L$

Theorem 9 *The local search Algorithm 2 returns a k^{-1}-approximation ratio of SPCA, i.e., the output $\widehat{S}_L$ of the local search Algorithm 2 satisfies*

$$\lambda_{\max}\Big(\sum_{i\in\widehat{S}_L}\boldsymbol{c}_i\boldsymbol{c}_i^\top\Big)\geq\frac{1}{k}w^*.$$

Proof. First, for each $j\in[n]$, we will show that

$$\lambda_{\max}\Big(\sum_{\ell\in\widehat{S}_L}\boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\Big)\geq\lambda_{\max}(\boldsymbol{c}_j\boldsymbol{c}_j^\top). \tag{24}$$

To prove it, there are two cases to be discussed: whether j belongs to $\widehat{S}_L$ or not. The monotonicity of the largest eigenvalue of sum of positive semi-definite matrices implies that the inequality (24) holds if $j\in\widehat{S}_L$. If $j\in[n]\setminus\widehat{S}_L$, then the local optimality condition implies that there exist $i\in\widehat{S}_L$ such that

$$\lambda_{\max}\Big(\sum_{\ell\in\widehat{S}_L}\boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\Big)\geq\lambda_{\max}\Big(\sum_{\ell\in\widehat{S}_L\cup\{j\}\setminus\{i\}}\boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\Big)\geq\lambda_{\max}(\boldsymbol{c}_j\boldsymbol{c}_j^\top),$$

where the second inequality is due to the monotonicity of the largest eigenvalue of sum of positive semi-definite matrices.

Second, suppose S^* to be the optimal solution to SPCA (4), by inequality (24), then we have

$$w^*=\lambda_{\max}\Big(\sum_{i\in S^*}\boldsymbol{c}_i\boldsymbol{c}_i^\top\Big)\leq\sum_{i\in S^*}\lambda_{\max}(\boldsymbol{c}_i\boldsymbol{c}_i^\top)\leq k\lambda_{\max}\Big(\sum_{\ell\in\widehat{S}_L}\boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\Big),$$

where the first inequality is because of the convexity of function $\lambda_{\max}(\cdot)$. □

We remark that Example 1 also confirms the tightness of our analysis for local search Algorithm 2.

Proposition 6 *In Example 1, the output value of local search Algorithm 2 is k^{-1}-away from optimal value of SPCA. That is, approximation ratio k^{-1} of local search Algorithm 2 is tight.*

Proof. In Example 1, we show that the initial subset $\widehat{S}_L=[k]$ already satisfies the local optimality condition.

Indeed, for each pair $(i,j)\in\widehat{S}_L\times([n]\setminus\widehat{S}_L)$, we have

$$\lambda_{\max}\Big(\sum_{\ell\in\widehat{S}_L\cup\{j\}\setminus\{i\}}\boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\Big)=\lambda_{\max}(\boldsymbol{I}_d-\boldsymbol{e}_i\boldsymbol{e}_i^\top)=1=\lambda_{\max}(\boldsymbol{I}_d-\boldsymbol{e}_d\boldsymbol{e}_d^\top)=\lambda_{\max}\Big(\sum_{\ell\in\widehat{S}_L}\boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\Big),$$

where the identities follow the construction of $\{\boldsymbol{c}_i\}_{i\in[n]}$ in Example 1.

Therefore, the set $\widehat{S}_L$ achieves the local optimum with largest eigenvalue of 1. Since the optimal value of SPCA is $w^*=k$, the approximation ratio of set $\widehat{S}_L$ is equal to k^{-1}. □

As an improved heuristic, local search Algorithm 2 can use the output of the greedy Algorithm 1 as an initial solution. The results in Theorem 9 and Proposition 6 imply that the integrated algorithm still yields a k^{-1}-approximation ratio of SPCA, while for solving the practical instances, our numerical study shows that the integrated algorithm in fact works very well. Since the greedy Algorithm 1 and local search Algorithm 2 repeatedly require to compute the largest eigenvalues, at each iteration, we can apply the power iteration method to efficiently calculate the largest eigenvalues [35] and use the eigenvectors from the previous iterations as a warm-start.

Finally, we remark that there is only one swap in the local search Algorithm 2. We can improve it by increasing the number of swapping elements at each iteration, termed *s-swap local search* with $s \in [k]$. The following result shows that s-swap local search can indeed achieve a better approximation ratio.

Corollary 2 *The approximation ratio of s-swap local search is sk^{-1} for any $s \in [k]$. The approximation ratio is tight.*

Proof. First, let set $\widehat{S}_L$ denote the indices of selected vectors by s-swap local search algorithm. Then following the same proof as that in Theorem 9, for any size-s set $T \subseteq [n]$, we have

$$\lambda_{\max}\Big(\sum_{i\in\widehat{S}_L} \boldsymbol{c}_i\boldsymbol{c}_i^\top\Big) \geq \lambda_{\max}\Big(\sum_{i\in T} \boldsymbol{c}_i\boldsymbol{c}_i^\top\Big). \tag{25}$$

Let S^* denote the optimal solution to SPCA (4), using the result (25), the optimal value of SPCA w^* is upper bounded by

$$w^* = \lambda_{\max}\Big(\sum_{i\in S^*} \boldsymbol{c}_i\boldsymbol{c}_i^\top\Big) = \lambda_{\max}\Big(\frac{1}{\binom{k-1}{s-1}}\sum_{T\subseteq S^*,|T|=s}\sum_{i\in T} \boldsymbol{c}_i\boldsymbol{c}_i^\top\Big) \leq \frac{\binom{k}{s}}{\binom{k-1}{s-1}}\lambda_{\max}\Big(\sum_{i\in\widehat{S}_L} \boldsymbol{c}_i\boldsymbol{c}_i^\top\Big) = \frac{k}{s}\Big(\sum_{i\in\widehat{S}_L} \boldsymbol{c}_i\boldsymbol{c}_i^\top\Big).$$

Second, to show the tightness, let us consider the following example.

Example 2 *For any integer $k \in [d]$, let $d = k+1$, $n = (s+1)k$, and the vectors $\{\boldsymbol{c}_i\}_{i\in[n]} \subseteq \mathbb{R}^d$ be*

$$\boldsymbol{c}_i = \begin{cases} \boldsymbol{e}_i, & \text{if } i \in [k], \\ \vdots & \\ \boldsymbol{e}_{i-(s-1)k}, & \text{if } i \in [(s-1)k+1, sk], \\ \boldsymbol{e}_{k+1}, & \text{if } i \in [sk+1, n], \end{cases} \quad \forall i \in [n].$$

In Example 2, we show that the subset $\widehat{S}_L = [k-s+1] \cup \{\ell k+1\}_{\ell\in[s-1]}$ satisfies the s-swap local optimality condition.

Indeed, for each pair (T_1, T_2) such that $T_1 \subseteq \widehat{S}_L, T_2 \subseteq ([n] \setminus \widehat{S}_L)$ with $|T_1| = |T_2| = s$, we have

$$\lambda_{\max}\Big(\sum_{\ell\in\widehat{S}_L\cup T_2\setminus T_1} \boldsymbol{c}_\ell\boldsymbol{c}_\ell^\top\Big) \leq s.$$

Therefore, the set $\widehat{S}_L$ achieves s-swap local optimum with largest eigenvalue of s. Since the optimal value of SPCA is $w^* = k$, the approximation ratio of set $\widehat{S}_L$ is equal to sk^{-1} for SPCA. □

Albeit theoretically sound, s-swap local search with $s \geq 2$ might not be practical since it involves $O(n^2)$ swaps at each iteration. Therefore, in the numerical study, we use the simple local search Algorithm 2, which already works very well.

6. Numerical Study In this section, we conduct numerical experiments on six datasets with number of features n ranging from 13 to 2365 to demonstrate the computational efficiency and the solution quality of the MISDP (6), MISDP (15), and MILP (22) for exactly solving SPCA, the continuous relaxations (8), (16) and heuristic Algorithms 1, 2 for approximately solving SPCA. All the methods in this section are coded in Python 3.6 with calls to Gurobi 9.0 and MOSEK 9.0 on a personal PC with 2.3 GHz Intel Core i5 processor and 8G of memory. The codes and data are available at `https://github.com/yongchunli-13/Sparse-PCA`.

6.1. *Pitprops* Dataset We first test the proposed three exact SPCA formulations (6), (15), (22) and their continuous relaxations to solve a commonly-used benchmark instance, *Pitprops* dataset Jeffers [20], which consists of 13 features (i.e., $n = 13$). In this instance, the computational results of seven different cases with k chosen from $\{4, \cdots, 10\}$ are displayed in Table 2, Table 3, and Table 4.

For each testing case, we solve two MISDP formulations (6) and (15) using the branch and cut method. As for the MILP (22), it can be simply solved in Gurobi. Throughout the numerical study of MILP (22), we set $\epsilon = 10^{-4}$, use the best SDP relaxation values as the upper bound w_U, and use the local search Algorithm 2 to compute the lower bound w_L. As the newly released Gurobi 9.0 is able to solve the non-convex quadratic program, thus for the purpose of comparison, we further use Gurobi to solve the following SPCA formulation

$$w^* := \max_{\boldsymbol{z} \in Z, \boldsymbol{x} \in \mathbb{R}^n} \left\{ \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x} : ||\boldsymbol{x}||_2 = 1, ||\boldsymbol{x}||_1 \leq \sqrt{k}, |x_i| \leq z_i, \forall i \in [n] \right\}. \tag{26}$$

The computation results of the exact methods are shown in Table 2. In particular, we let **time(s)** denote the running time in seconds of each case and let **Gurobi** denote the performance of Gurobi for solving SPCA (26). In table 2, we see that all the SPCA formulations (6), (15), (22) can be solved to optimality within seconds, which demonstrates the efficiency of the proposed formulations. We also compare the numerical performance of the MILP formulation (22) with formulation (26) using the Gurobi solver, and it is clear that MILP is more efficient and stable. Especially for the case of $k = 10$, Gurobi has trouble finding the optimal solution of SPCA (26).

Although the theoretical optimality gaps of the proposed SDP relaxations (8) and (16) are the same, these gaps in practice can be much smaller and can be significantly different from each other.

TABLE 2. Computational results of exact values with *Pitprops* dataset

n=13	SPCA	MISDP (6)		MISDP (15)		MILP (22)		Gurobi	
k	w^*	w^*	time(s)	w^*	time(s)	$\widehat{w}(\epsilon)$	time(s)	w^*	time(s)
4	2.9375	2.9375	1	2.9375	2	2.9375	1	2.9375	1
5	3.4062	3.4062	1	3.4062	2	3.4062	1	3.4062	1
6	3.7710	3.7710	1	3.7710	2	3.7710	2	3.7710	1
7	3.9962	3.9962	1	3.9962	1	3.9962	1	3.9962	3
8	4.0686	4.0686	1	4.0686	2	4.0686	2	4.0686	12
9	4.1386	4.1386	1	4.1386	2	4.1386	1	4.1387	30
10	4.1726	4.1726	1	4.1726	1	4.1726	1	4.1441	83

We use MOSEK to solve both SDP relaxations. The numerical results can be found in Table 3, where the SDP relaxation (17) proposed by d'Aspremont et al. [13] is presented as a benchmark comparison. In Table 3, we use **gap(%)** to denote the optimality gap, which is computed as $100 \times (\text{Upper Bound} - w^*)/w^*$. It can be seen that the second SDP relaxation (16) is superior to the first SDP relaxation (8) on the first five cases. When k is close to n, the first SDP relaxation (8) can be better. This finding is consistent with remarks after Theorem 2. In addition, as proved in Proposition 3, we see that the second SDP relaxation (16) always outperforms the bound (17) by d'Aspremont et al. [13]. Finally, the second SDP relaxation (16) and the bound (17) by d'Aspremont et al. [13] are also not comparable.

TABLE 3. Computational results of upper bounds with *Pitprops* dataset

n=13	SPCA	Benchmark (17)		SDP Relaxation (8)			SDP Relaxation (16)		
k	w^*	$\overline{w}_4$	gap(%)	$\overline{w}_1$	gap(%)	time(s)	$\overline{w}_3$	gap(%)	time(s)
4	2.9375	3.0172	2.71	3.1065	5.75	0.51	2.9495	0.41	0.13
5	3.4062	3.4581	1.52	3.4868	2.37	0.55	3.4124	0.18	0.18
6	3.7710	3.8137	1.13	3.7859	0.39	0.52	3.7767	0.15	0.15
7	3.9962	4.0316	0.89	3.9962	0.00	0.43	3.9962	0.00	0.15
8	4.0686	4.1448	1.87	4.0805	0.29	0.29	4.0793	0.26	0.17
9	4.1386	4.2063	1.64	4.1386	0.00	0.00	4.1398	0.03	0.15
10	4.1726	4.2186	1.10	4.1763	0.09	0.09	4.1778	0.12	0.16

Table 4 presents the objective values and optimality gaps of the proposed approximation algorithms for solving the *Pitprops* instance, where we let **LB** denote the lower bound and compute **gap(%)** by $100 \times (w^* - \text{LB})/w^*$. Note that we initialize the local search Algorithm 2 by the output of greedy Algorithm 1. To further improve the two algorithms, at each iteration, we employ the power iteration method to efficiently compute the largest eigenvalues [35] and warm-start it with

the good-quality eigenvectors from the previous iterations. In Table 4, we see that greedy Algorithm 1 and local search Algorithm 2 successfully find the optimal solutions and outperforms the truncation algorithm proposed by [9].

TABLE 4. Computational results of lower bounds with *Pitprops* dataset

n=13	SPCA	Truncation algorithm [9]			Greedy Algorithm 1			Local Search Algorithm 2		
k	w^*	LB	gap(%)	time(s)	LB	gap(%)	time(s)	LB	gap(%)	time(s)
4	2.9375	2.8913	1.57	1e-3	2.9375	0.00	1e-3	2.9375	0.00	1e-2
5	3.4062	3.3951	0.32	1e-3	3.4062	0.00	1e-3	3.4062	0.00	1e-2
6	3.7710	3.7576	0.36	1e-3	3.7710	0.00	1e-2	3.7710	0.00	1e-2
7	3.9962	3.9929	0.08	1e-3	3.9962	0.00	1e-2	3.9962	0.00	1e-2
8	4.0686	4.0648	0.09	1e-3	4.0686	0.00	1e-2	4.0686	0.00	1e-2
9	4.1386	4.1313	0.18	1e-3	4.1386	0.00	1e-2	4.1386	0.00	1e-2
10	4.1726	4.0094	3.91	1e-3	4.1726	0.00	1e-2	4.1726	0.00	1e-2

6.2. Four Larger-scale Datasets In this subsection, we conduct experiments on four larger instances from Dey et al. [15] to further testify the efficiency of our proposed methods for SPCA, which are *Eisen-1*, *Eisen-2*, *Colon* and *Reddit* with n =79, 118, 500, and 2000. Since the MILP formulation (22) consistently outperforms two MISDP formulations (6) and (15). Thus, in this set of numerical experiments, we will stick to the MILP formulation (22).

We first compare the performances of different heuristic methods using the *Reddit* dataset with $n = 2000$ and $k \in \{10, \ldots, 70\}$. Thus, there are 7 cases in total. We implement the greedy Algorithm 1 and the local search Algorithm 2 and compare them with the best-known truncation algorithm proposed by [9]. The numerical results are shown in Table 5. We see that the local search Algorithm 2 provides the highest-quality solution of the three. The greedy Algorithm 1 is almost equally as good as the truncation algorithm. Although the local search Algorithm 2 takes the longest running time, the running time is quite reasonable given the size of the testing cases. Hence, our computation experiments show that the local search Algorithm 2 consistently outperforms the other two methods within a reasonably short time. Thus, we recommend using this algorithm to solve practical problems.

Next, we obtain the local search Algorithm 2, the continuous relaxation bounds and exact values of SPCA on the four instances, i.e., *Eisen-1*, *Eisen-2*, *Colon* and *Reddit*. For these instances, MOSEK fails to solve our proposed SDP relaxations (8) and (16). Thus, instead, we use the subgradient method to solve the continuous relaxation formulations (13) and (20). For the MILP formulation (22), we set the time limit of Gurobi to be an hour. The computational results are presented in Table 6, where we let **UB** denote the upper bound of SPCA, let **VAL** denote the

TABLE 5. Computational results of lower bounds with *Reddit* dataset

n=2000	Truncation algorithm [9]		Greedy Algorithm 1		Local Search Algorithm 2	
k	LB	time (s)	LB	time (s)	LB	time (s)
10	1482.3205	3	1521.3081	1	1521.3083	9
20	1666.2397	2	1670.4712	4	1684.3943	59
30	1953.3711	2	1856.2875	7	1953.7502	92
40	2203.1715	2	2123.5635	10	2208.2452	208
50	2311.2407	2	2289.0371	13	2322.8204	207
60	2427.2685	3	2402.8345	16	2441.7020	202
70	2475.9581	2	2488.8991	19	2494.6142	193

best lower bound of MILP (22) found if the time limit is reached, and let **MIPgap(%)** denote the percentage of output MIP Gap from Gurobi. For these instances, we see that the local search Algorithm 2 still performs very well and the subgradient method is also efficient to solve the continuous relaxation (13). The continuous relaxation (20) turns out to be very difficult to compute, and even more difficult than the MILP formulation (22). For the instance *Eisen-1*, we see that both the MILP formulation (22) and local search Algorithm 2 can find the optimal solutions. This further demonstrates the effectiveness of the local search Algorithm 2.

TABLE 6. Computational results of lower bounds, upper bounds and exact values with four larger instances

Data	Case		Local Search Algorithm 2		Continuous Relaxation (13)		Continuous Relaxation (20)		MILP (22)		
	n	k	LB	time(s)	UB	time(s)	UB	time(s)	VAL	MIPgap(%)	time(s)
Eisen-1	79	10	17.3355	1	17.9144	14	17.7571	126	17.3355	0.00	34
	79	20	17.7195	1	18.1309	13	18.0362	85	17.7195	0.00	125
Eisen-2	118	10	11.7182	1	13.8732	89	-	-	11.7182	18.39	3600
	118	20	19.3228	1	22.9268	90	-	-	19.3228	18.65	3600
Colon	500	10	2641.2289	1	2901.1105	342	-	-	2641.2289	9.84	3600
	500	20	4255.6941	3	4833.1900	344	-	-	4255.6941	13.57	3600
Reddit	2000	10	1521.3083	9	1867.9965	1198	-	-	-	-	-
	2000	20	1684.3943	59	2184.2436	1241	-	-	-	-	-

6.3. *Drugabuse* Dataset We finally apply the proposed local search Algorithm 2 to the *Drugabuse* Dataset with $n = 2365$ features, where the dataset comes from a questionnaire collected by the National Survey on Drug Use and Health (NSDUH) in 2018. It has been reported [33] that with the growing illicit online sale of controlled substances, deaths attributable to opioid-related drugs have been more than quadrupled in the U.S. since 1999. Thus, it is important to select a handful of features that the researchers can focus on for further exploration. Indeed, SPCA is a good tool to reduce the complexity and improve the interpretability of the machine learning algorithms

by selecting the most important features. Our numerical finding of the case of $k = 10$ is illustrated in Figure 1, where the vertical values correspond to the selected features of the first PC, which are scaled by 100. We see that among 10 features, there are three categories (i.e., inhalants, drug injection, drug treatment), which are important for analyzing drug abuse. In particular, SPCA selects 6 features related to drug treatment, which is consistent with the literature [11, 39] that the treatment records of drug abuse are informative and important. Three drug injection questions have been designed to understand the injection experience of different special drugs, and it is well known that drug injection users are at high risk for HIV and other blood-borne infections [32, 38]. Inhalants feature, corresponding to various accessible products that can easily cause addictions, significantly contributes to the increase of drug abuse [6, 14].

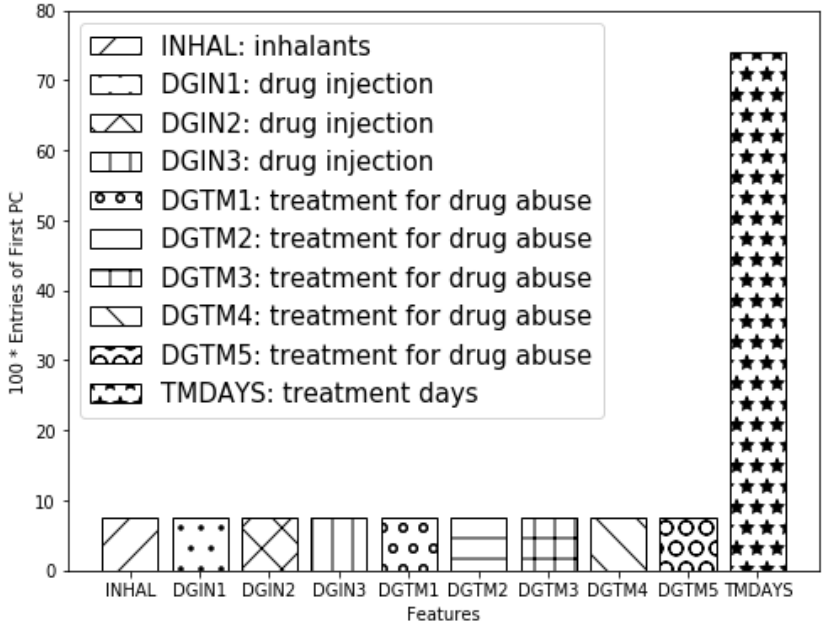

FIGURE 1. 10 features selected by local search Algorithm 2 for *Drugabuse* dataset

7. Extension to the Rank-one Sparse Singular Value Decomposition (R1-SSVD) In this section, we extend the proposed formulations and theoretical results to the rank-one sparse singular value decomposition (R1-SSVD). R1-SSVD has been successfully used to analyze the row-column associations within high-dimensional data (see, e.g., [28, 23, 36]). The goal of R1-SSVD is to find the best submatrix (possibly non-square) of a particular size whose largest singular value is maximized, from a given matrix.

Formally, R1-SSVD can be formulated as

$$\text{(R1-SSVD)} \quad w^*_{\text{SVD}} := \max_{\boldsymbol{u}\in\mathbb{R}^m, \boldsymbol{v}\in\mathbb{R}^n} \left\{ \boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v} : ||\boldsymbol{u}||_2 = 1, ||\boldsymbol{v}||_2 = 1, ||\boldsymbol{u}||_0 = k_1, ||\boldsymbol{v}||_0 = k_2 \right\}, \tag{27}$$

where the matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is known, m, n, and $k_1 \in [m]$ and $k_2 \in [n]$ are positive integers.

Our reduction of R1-SSVD (27) to SPCA (1) follows from the development of an augmented symmetric matrix $\overline{\boldsymbol{A}} \in \mathcal{S}^{m+n}$

$$\overline{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{0} \end{bmatrix}. \tag{28}$$

Let $\boldsymbol{x} := [\boldsymbol{u}^\top, \boldsymbol{v}^\top]^\top$ denote an $(m+n)$-dimensional vector. According to the identity

$$\boldsymbol{x}^\top \overline{\boldsymbol{A}} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{u}^\top & \boldsymbol{v}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} \\ \boldsymbol{A}^\top & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} = 2\boldsymbol{u}^\top \boldsymbol{A} \boldsymbol{v},$$

then R1-SSVD (27) can be reformulated as

$$w^*_{\mathrm{SVD}} := \frac{1}{2} \max_{\boldsymbol{x}\in\mathbb{R}^{m+n}} \left\{ \boldsymbol{x}^\top \overline{\boldsymbol{A}} \boldsymbol{x} : ||\boldsymbol{x}_{1:m}||_2 = 1, ||\boldsymbol{x}_{m+1:m+n}||_2 = 1, ||\boldsymbol{x}_{1:m}||_0 = k_1, ||\boldsymbol{x}_{m+1:m+n}||_0 = k_2 \right\}, \tag{29}$$

where we let $\boldsymbol{x}_{1:m}$ denote the collection of m entries of vector $\boldsymbol{x}$ from index set $[m]$ and $\boldsymbol{x}_{m+1,m+n}$ denote the n entries of $\boldsymbol{x}$ from index set $[m+1, m+n]$. In R1-SSVD (29), we enforce the sparse restrictions on both $\boldsymbol{x}_{1:m}$ and $\boldsymbol{x}_{m+1,m+n}$. Thus, the R1-SSVD (29) can be viewed as a special case of the conventional SPCA (1), where $\boldsymbol{A}$ is symmetric but not positive semi-definite and there are two sparsity constraints instead of one.

Similarly, introducing binary variable $z_i = 1$ if ith column of matrix $\overline{\boldsymbol{A}}$ is chosen, 0, otherwise, we can linearize the zero-norm constraints and recast R1-SSVD (29) as

$$w^*_{\mathrm{SVD}} := \frac{1}{2} \max_{\boldsymbol{x}\in\mathbb{R}^{m+n}, \boldsymbol{z}\in Z_{\mathrm{SVD}}} \left\{ \boldsymbol{x}^\top \overline{\boldsymbol{A}} \boldsymbol{x} : ||\boldsymbol{x}_{1:m}||_2 = 1, ||\boldsymbol{x}_{m+1:m+n}||_2 = 1, |x|_i \le z_i, \forall i \in [m+n] \right\}, \tag{30}$$

where set Z_{SVD} is defined as

$$Z_{\mathrm{SVD}} := \left\{ \boldsymbol{z} \in \{0,1\}^{m+n} : \sum_{i\in[m]} z_i = k_1, \sum_{i\in[m+1,m+n]} z_i = k_2 \right\}.$$

The following lemma inspires us three exact mixed-integer formulations for R1-SSVD (30).

Lemma 3 *Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, consider its augmented counterpart $\overline{\boldsymbol{A}}$ defined in (28), two integers $k_1 \in [m]$ and $k_2 \in [n]$, and three subsets $S, S_1, S_2 \subseteq [m+n]$ such that $S \subseteq [m+n], |S| = k_1 + k_2$, $S_1 = S \cap [m], |S_1| = k_1$ and $S_2 = S \cap [m+1, m+n], |S_2| = k_2$. Then the following identities must hold:*

(i) The eigenvalues of the augmented submatrix $\overline{\boldsymbol{A}}_{S,S}$ are the singular values of submatrix $\boldsymbol{A}_{S_1,S_2}$ and their negations;

(ii) $\sigma_{\max}(\boldsymbol{A}_{S_1,S_2}) = \lambda_{\max}(\overline{\boldsymbol{A}}_{S,S}) = 1/2 \max_{\boldsymbol{x}\in\mathbb{R}^{k_1+k_2}} \{\boldsymbol{x}^\top \overline{\boldsymbol{A}} \boldsymbol{x} : ||\boldsymbol{x}_{1:k_1}||_2 = 1, ||\boldsymbol{x}_{k_1+1:k_1+k_2}||_2 = 1\} = 1/2 \max_{\boldsymbol{X}\in\mathcal{S}^{k_1+k_2}_+} \left\{ \mathrm{tr}(\overline{\boldsymbol{A}}_{S,S}\boldsymbol{X}) : \sum_{j\in[k_1]} X_{jj} = 1, \sum_{i\in[k_1+1,k_1+k_2]} X_{ii} = 1 \right\}$.

Proof. See Appendix A.7. □

Notably, Part (ii) in Lemma 3 shows that R1-SSVD is equivalent to the following combinatorial optimization problem

$$w^*_{\rm SVD} := \max_{S\subseteq[m+n]} \left\{ \lambda_{\max}(\overline{\boldsymbol{A}}_{S,S}) : |S\cap[m]| = k_1, |S\cap[m+1,m+n]| = k_2 \right\}. \tag{31}$$

The next four subsections present MISDP formulations (I) and (II), a MILP formulation, and approximation algorithms, respectively.

7.1. MISDP Formulation (I) The fact that matrix $\overline{\boldsymbol{A}}$ is symmetric but not positive semi-definite impedes us to directly apply the results in Section 2. Fortunately, a simple remedy by adding a new matrix $\sigma_{\max}(\boldsymbol{A})\boldsymbol{I}_{m+n}$ to $\overline{\boldsymbol{A}}$ fixes this issue. That is, let us define

$$\overline{\boldsymbol{A}}^{\#} := \overline{\boldsymbol{A}} + \sigma_{\max}(\boldsymbol{A})\boldsymbol{I}_{m+n}, \tag{32}$$

which is indeed positive semi-definite according to Part (i) in Lemma 3. More importantly, the new matrix $\overline{\boldsymbol{A}}^{\#}$ preserves all the sparsity properties of the original one $\overline{\boldsymbol{A}}$.

Thus, the combinatorial optimization R1-SSVD (30) is equivalent to

$$w^*_{\rm SVD} := \max_{S\subseteq[m+n]} \left\{ \lambda_{\max}(\overline{\boldsymbol{A}}^{\#}_{S,S}) : |S\cap[m] = k_1, |S\cap[m+1,m+n] = k_2 \right\} - \sigma_{\max}(\boldsymbol{A}). \tag{33}$$

Now all the results in Section 2 are directly applicable to R1-SSVD (33). We highlight two important ones below.

Theorem 10 *The R1-SSVD* (33) *admits an equivalent MISDP formulation:*

$$w^*_{\rm SVD} := \max_{\substack{\boldsymbol{z}\in Z_{\rm SVD},\\ \boldsymbol{X},\boldsymbol{W}_1,\cdots,\boldsymbol{W}_d\in\mathcal{S}^d_+}} \left\{ \sum_{i\in[m+n]} \boldsymbol{c}_i^\top \boldsymbol{W}_i \boldsymbol{c}_i : {\rm tr}(\boldsymbol{X}) = 1, \boldsymbol{X} \succeq \boldsymbol{W}_i, {\rm tr}(\boldsymbol{W}_i) = z_i, \forall i\in[m+n] \right\} - \sigma_{\max}(\boldsymbol{A}), \tag{34}$$

where $\overline{\boldsymbol{A}}^{\#} = \boldsymbol{C}^\top\boldsymbol{C}$ *denotes the Cholesky factorization of* $\overline{\boldsymbol{A}}^{\#}$ *with* $\boldsymbol{C}\in\mathbb{R}^{d\times(m+n)}$, d *is the rank of* $\overline{\boldsymbol{A}}^{\#}$, *and* $\boldsymbol{c}_i\in\mathbb{R}^d$ *denotes the* i*-th column vector of matrix* $\boldsymbol{C}$ *for each* $i\in[m+n]$.

Theorem 11 *The continuous relaxation value* $\overline{w}_{\rm SVD1}$ *of formulation* (34) *satisfies*

$$w^*_{\rm SVD} \le \overline{w}_{\rm SVD1} \le \sqrt{mnk_1^{-1}k_2^{-1}}\, w^*_{\rm SVD}.$$

Proof. See Appendix A.8. □

7.2. MISDP Formulation (II) Since the results in Section 3 do not rely on the positive semi-definiteness of matrix $\boldsymbol{A}$, they can be directly extended to R1-SSVD (30).

We first illustrate a naive MISDP for R1-SSVD (30) based on Part (ii) in Lemma 3.

Proposition 7 *The R1-SSVD* (30) *is equivalent to the following MISDP formulation:*

$$w^*_{\rm SVD} := \frac{1}{2} \max_{\boldsymbol{z}\in Z_{\rm SVD}, \boldsymbol{X}\in\mathcal{S}^{m+n}_+} \left\{ {\rm tr}(\overline{\boldsymbol{A}}\boldsymbol{X}) : \sum_{j\in[m]} X_{jj} = 1, \sum_{j\in[m+1,m+n]} X_{jj} = 1, X_{ii} \le z_i, \forall i\in[m+n] \right\}. \tag{35}$$

The R1-SSVD formulation (35) is rather weak and its continuous relaxation value is equal to $\sigma_{\max}(\boldsymbol{A})$. Fortunately, we can derive two types of valid inequalities from strengthening it as below.

Lemma 4 *For R1-SSVD* (35)*, the following second-order conic inequalities are valid:*

(i) $\sum_{j\in[m]} X^2_{ij} \le z_i X_{ii}, \sum_{j\in[m+1,m+n]} X^2_{ij} \le z_i X_{ii}$ *for all* $i\in[m+n]$*; and*

(ii) $(\sum_{j\in[m]} |X_{ij}|)^2 \le k_1 X_{ii} z_i, (\sum_{j\in[m+1,m+n]} |X_{ij}|)^2 \le k_2 X_{ii} z_i$ *for all* $i\in[m+n]$.

Proof. See Appendix A.9. □

The MISDP formulation for R1-SSVD (35) can be strengthened by adding these valid inequalities. Similar to Theorem 5, we provide the optimality gap of its continuous relaxation value as below.

Theorem 12 *The continuous relaxation value* $\overline{w}_{\rm SVD2}$ *of R1-SSVD* (35) *with the inequalities in Lemma 4 yields an optimality gap at most* $\min\{\sqrt{k_1k_2}, mnk_1^{-1}k_2^{-1}\}$*, i.e.,*

$$w^*_{\rm SVD} \le \overline{w}_{\rm SVD2} \le \min\left\{\sqrt{k_1k_2}, \sqrt{mnk_1^{-1}k_2^{-1}}\right\} w^*_{\rm SVD}.$$

7.3. An MILP Formulation with Arbitrary Accuracy Similarly, we can develop an MILP formulation with arbitrary accuracy based on the Cholesky decomposition of matrix $\overline{\boldsymbol{A}}^{\#}$ in R1-SSVD (33). The proofs are similar to Section 4 and are thus omitted.

Theorem 13 *Given a threshold* $\epsilon > 0$ *and lower and upper bounds of the optimal R1-SSVD,* w_L, w_U*, the following MILP is* $O(\epsilon)$*-approximate to R1-SSVD* (33)*, i.e.,* $\epsilon \le \widehat{w}(\epsilon) - w^* \le \epsilon\sqrt{d}$*:*

$$\begin{aligned}
\widehat{w}(\epsilon) := \max_{w,\boldsymbol{z}\in Z_{\rm SVD},\boldsymbol{y},\boldsymbol{\alpha},\boldsymbol{x},,\boldsymbol{\delta},\boldsymbol{\mu},\boldsymbol{\sigma}} & w - \sigma_{\max}(\boldsymbol{A}) \\
\text{s.t.} \quad & \boldsymbol{x} = \boldsymbol{\delta}_{i1} + \boldsymbol{\delta}_{i2}, ||\boldsymbol{\delta}_{i1}||_\infty \le z_i, ||\boldsymbol{\delta}_{i2}||_\infty \le 1 - z_i, \forall i\in[m+n], \\
& \boldsymbol{x} = \sum_{j\in[d]} \boldsymbol{\sigma}_j, ||\boldsymbol{\sigma}_j||_\infty \le y_j, \sigma_{jj} = y_j, \forall j\in[d], \sum_{j\in[d]} y_j = 1, \\
& \boldsymbol{x} = \boldsymbol{\mu}_{\ell 1} + \boldsymbol{\mu}_{\ell 2}, ||\boldsymbol{\mu}_{\ell 1}||_\infty \le \alpha_\ell, ||\boldsymbol{\mu}_{\ell 2}||_\infty \le 1 - \alpha_\ell, \forall \ell\in[L], \\
& w = w_U - (w_U - w_L)\left(\sum_{i\in[L]} 2^{-i}\alpha_i\right), \\
& \left|\left| \sum_{i\in[m+n]} \boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1} - w_U\boldsymbol{x} + (w_U - w_L)\sum_{i\in[L]} 2^{-i}\boldsymbol{\mu}_{i1} \right|\right|_\infty \le \epsilon, \\
& \boldsymbol{\alpha}\in\{0,1\}^L, \boldsymbol{y}\in\{0,1\}^d,
\end{aligned} \tag{36}$$

where $L := \lceil \log_2(\epsilon/(w_U - w_L)) \rceil$.

Theorem 14 *Given a threshold* $\epsilon > 0$*, let* $\overline{w}_{\mathrm{SVD3}}(\epsilon)$ *denote the optimal value of MILP formulation* (36) *by relaxing the binary variables* $\boldsymbol{z}$ *to be continuous. Then we have*

$$\overline{w}_{\mathrm{SVD3}}(\epsilon) \leq \sqrt{\frac{mn}{k_1 k_2}} \left[\min \left\{ (k_1+k_2)\frac{\sqrt{d}+1}{2}, \right. \right. \\ \left. \left. \frac{m+n}{k_1+k_2}\sqrt{d} + (m+n-k_1-k_2)\frac{\sqrt{d}+1}{2} \right\} - 1 \right] w^*_{\mathrm{SVD}} + \epsilon\sqrt{d}.$$

7.4. Approximation Algorithms for R1-SSVD We will investigate three approximation algorithms for R1-SSVD (27): truncation algorithm, greedy algorithm, and local search algorithm.

7.4.1. Truncation algorithm The approximation algorithm in [9] via truncation is known so far with the best approximation ratio $O(n^{-1/3})$ for SPCA. We show that a similar truncation also works for R1-SSVD.

First, we define the truncation operator as below.

Definition 1 (Normalized Truncation) *Given a vector* $\boldsymbol{x} \in \mathbb{R}^n$ *and an integer* $s \in [n]$*, vector* $\widehat{\boldsymbol{x}}$ *is an* s*-truncation of* $\boldsymbol{x}$ *if*

$$\widehat{x}_i = \begin{cases} |x_i|, & \text{if } |x_i| \text{ is one of the } s \text{ largest absolute entries of } \boldsymbol{x} \\ 0, & \text{otherwise} \end{cases}$$

for each $i \in [n]$*. The normalized* s*-truncation of* $\boldsymbol{x}$ *is defined as* $\widehat{\boldsymbol{x}} := \widehat{\boldsymbol{x}}/\|\widehat{\boldsymbol{x}}\|_2$*, which is normalized to be of unit length.*

Then the truncation algorithm for R1-SSVD has the following two steps:

(i) Truncation in the standard basis: For each $i \in [n]$, let $\widehat{\boldsymbol{u}}_i \in \mathbb{R}^m$ be the normalized k_1-truncation on the i-th column vector of $\boldsymbol{A}$, and for each $j \in [m]$, let $\widehat{\boldsymbol{v}}_j \in \mathbb{R}^n$ be the normalized k_2-truncation on the j-th row vector of $\boldsymbol{A}$. Clearly, $\widehat{\boldsymbol{u}}_i$ and $\widehat{\boldsymbol{v}}_j$ are feasible to R1-SSVD (27);

(ii) Truncation in the eigen-space basis: Let $\boldsymbol{v}_1$ and $\boldsymbol{u}_1$ denote the right and left eigenvectors of $\boldsymbol{A}$ corresponding to the largest singular value. We then define the vector $\widehat{\boldsymbol{u}}_1$ as the normalized k_1-truncation on $\boldsymbol{u}_1$ and define $\widehat{\boldsymbol{v}}_1$ as the normalized k_2-truncation of the vector $\boldsymbol{A}^\top \widehat{\boldsymbol{u}}_1$. It is clear that $(\widehat{\boldsymbol{u}}_1, \widehat{\boldsymbol{v}}_1)$ is also feasible to R1-SSVD (27).

The approximation results of the truncation procedure are summarized below.

Theorem 15 *For R1-SSVD* (27)*, the truncation algorithm yields an approximation ratio*

$$\max\left\{ \sqrt{k_1^{-1}}, \sqrt{k_2^{-1}}, \sqrt{k_1 k_2 m^{-1} n^{-1}} \right\}.$$

In particular, the approximation ratio is $O(n^{-1/3})$ *when* $k_1 \approx k_2$ *and* $m \approx n$*.*

Proof. See Appendix A.10. □

7.4.2. Greedy and Local Search Algorithms We design the greedy and local search algorithms according to the following equivalent combinatorial formulation of R1-SSVD (27)

$$w^*_{\rm SVD} := \max_{S_1\subseteq[m],S_2\subseteq[n]}\left\{\sigma_{\max}(\boldsymbol{A}_{S_1,S_2}) : |S_1| = k_1, |S_2| = k_2\right\}. \tag{37}$$

Different from SPCA (3), the R1-SSVD (37) maximizes the largest singular value of any $k_1 \times k_2$ submatrix rather than that of any size k-principal submatrix. Therefore, to solve R1-SSVD (37), we adapt the greedy Algorithm 1 or the local search Algorithm 2 considering selecting a row and/or a column at each iteration.

Specifically, for the greedy algorithm, let two subsets S_1, S_2 denote the index sets of the selected columns and rows, respectively. We first initialize the greedy algorithm by selecting the entry of $\boldsymbol{A}$ that takes the largest absolute value. Then, we add one element into each subset at each iteration, which maximizes the largest singular value of the obtained submatrix, unless we are not able to. Next, we continue to selection one row (or one column) at each iteration, until we reach a $k_1 \times k_2$ submatrix. The detailed implementation can be found in Algorithm 3.

Given an initial feasible solution (S_1, S_2) to R1-SSVD (37), the adapted local search algorithm performs the swapping procedure on both S_1 and S_2 (see Algorithm 4 for details) simultaneously.

Algorithm 3 Greedy Algorithm for R1-SSVD (37)

Input: $m \times n$ matrix $\boldsymbol{A} \succeq 0$, integers $k_1 \in [m]$, $k_2 \in [n]$
Let $\widehat{S}_1 := \emptyset$ and $\widehat{S}_2 := \emptyset$ denote the selected rows and columns, separately
Compute $j_1^*, j_2^* \in \arg\max_{j_1\in[m],j_2\in[n]} \left\{|(\boldsymbol{A}_{\{j_1\},\{j_2\}}|\right\}$
Add j_1^*, j_2^* to sets $\widehat{S}_1$ and $\widehat{S}_2$, separately
for $\ell = 2, \cdots, \max\{k_1, k_2\}$ **do**
    if $\ell \le \min\{k_1, k_2\}$ **then**
        Compute $j_1^* \in \arg\max_{j_1\in[m]\setminus\widehat{S}_1} \left\{\sigma_{\max}\left(\boldsymbol{A}_{\widehat{S}_1\cup\{j_1\},\widehat{S}_2}\right)\right\}$ and add j_1^* to set $\widehat{S}_1$
        Compute $j_2^* \in \arg\max_{j_2\in[n]\setminus\widehat{S}_2} \left\{\sigma_{\max}\left(\boldsymbol{A}_{\widehat{S}_1,\widehat{S}_2\cup\{j_2\}}\right)\right\}$ and add j_2^* to set $\widehat{S}_2$
    else if $k_1 \le k_2$ **then**
        Compute $j_2^* \in \arg\max_{j_2\in[n]\setminus\widehat{S}_2} \left\{\sigma_{\max}\left(\boldsymbol{A}_{S_1,S_2\cup\{j_2\}}\right)\right\}$ and add j_2^* to set $\widehat{S}_2$
    else
        Compute $j_1^* \in \arg\max_{j_1\in[m]\setminus\widehat{S}_1} \left\{\sigma_{\max}\left(\boldsymbol{A}_{\widehat{S}_1\cup\{j_1\},\widehat{S}_2}\right)\right\}$ and add j_1^* to set $\widehat{S}_1$
    end if
end for
Output: $\widehat{S}_1, \widehat{S}_2$

The following results illustrate the theoretical performance guarantees of the two algorithms for R1-SSVD and show that the approximation ratios are both tight.

Theorem 16 *For the greedy Algorithm 3 and the local search Algorithm 4, we have (i) both algorithms achieve a $(\sqrt{k_1 k_2})^{-1}$-approximation ratio of R1-SSVD* (37)*, and (ii) the ratio is tight.*

Proof. See Appendix A.11. □

Algorithm 4 Local Search Algorithm for R1-SSVD (37)

Input: $m \times n$ matrix $\boldsymbol{A} \succeq 0$ and integers $k_1 \in [m]$, $k_2 \in [n]$
Initialize a size-k_1 subset $\widehat{S}_1 \subseteq [m]$ and a size-k_2 subset $\widehat{S}_2 \subseteq [n]$
do
  for each pair $(i_1, j_1, i_2, j_2) \in \widehat{S}_1 \times ([m] \setminus \widehat{S}_1) \times \widehat{S}_2 \times ([n] \setminus \widehat{S}_2)$ **do**
    if $\sigma_{\max}\left(\boldsymbol{A}_{S_1 \cup \{j_1\} \setminus \{i_1\}, S_2 \cup \{j_2\} \setminus \{i_2\}}\right) > \sigma_{\max}\left(\boldsymbol{A}_{S_1, S_2}\right)$ **then**
      Update $\widehat{S}_1 := \widehat{S}_1 \cup \{j_1\} \setminus \{i_1\}$, $\widehat{S}_2 := \widehat{S}_2 \cup \{j_2\} \setminus \{i_2\}$
    end if
  end for
while there is still an improvement
Output: $\widehat{S}_1$, $\widehat{S}_2$

8. Extension to Sparse Fair PCA In this section, we study the Sparse Fair PCA (SFPCA) and show its approximate MISDP formulation. The fair PCA has been recently studied in the literature (see, e.g., [34, 37]). The goal of SFPCA is to seek the best principal submatrices of multi-group covariance matrices to achieve the relatively similar objective values among different groups.

Suppose there are s groups and their corresponding covariance matrices are $\{\boldsymbol{A}_i\}_{i \in [s]}$. Then the SFPCA can be formulated as

$$w_F^* := \max_{\boldsymbol{x}} \left\{ \min_{i \in S} \boldsymbol{x}^\top \boldsymbol{A}_i \boldsymbol{x} : ||\boldsymbol{x}||_2 = 1, ||\boldsymbol{x}||_0 \leq k \right\}. \tag{38}$$

By introducing binary variables $\boldsymbol{z}$ and linearizing the objective function, we obtain

$$w_F^* := \max_{w, \boldsymbol{x}, \boldsymbol{z} \in Z} \left\{ w : w \leq \boldsymbol{x}^\top \boldsymbol{A}^i \boldsymbol{x}, \forall i \in [s], ||\boldsymbol{x}||_2 = 1, -z_i \leq x_i \leq z_i, \forall i \in [n] \right\}. \tag{39}$$

As the SFPCA (39) is quite different from SPCA, it is not surprising that the results in Section 2 and Section 4 do not apply to SFPCA (39). Fortunately, the results in Section 3 do provide an interesting upper bound for SFPCA (39), which can be exact when there are $s = 2$ groups of covariance matrices. Introducing a rank-one positive semi-definite matrix variable $\boldsymbol{X} \in \mathcal{S}_+^n$ such

that $\boldsymbol{X} \succeq \boldsymbol{x}\boldsymbol{x}^\top$, dropping the rank-one restriction, and adding the valid inequalities in Theorem 4, the problem (39) can be upper bounded by

$$\overline{w}_F := \max_{w,\boldsymbol{X},\boldsymbol{z}\in Z} \Bigg\{ w : w \le \mathrm{tr}(\boldsymbol{A}^i\boldsymbol{X}), \forall i \in [s], \mathrm{tr}(\boldsymbol{X}) = 1, \\ \sum_{j\in[n]} X_{ij}^2 \le X_{ii} z_i, \Big(\sum_{j\in[n]} |X_{ij}| \Big)^2 \le k X_{ii} z_i, \forall i \in [n] \Bigg\}. \tag{40}$$

The following result shows that if $s = 2$, then the approximation (40) is exact, otherwise, it provides an upper bound of SFPCA (39).

Proposition 8 *For the MISDP formulation* (40)*, we have*

(i) The optimal value of MISDP formulation (40) *provides an upper bound of SFPCA* (39)*, i.e.,* $\overline{w}_F \ge w_F^*$*. Also, when* $s = 2$*, the formulation* (40) *becomes exact, i.e.,* $\overline{w}_F = w_F^*$*; and*

(ii) There exists an optimal solution $(w^*, \boldsymbol{X}^*, \boldsymbol{z}^*)$ *of of MISDP* (40) *such that the rank of* $\boldsymbol{X}^*$ *is at most* $1 + \lfloor \sqrt{2s + 9/4} - 3/2 \rfloor$*.*

Proof.

(i) It is clear that $\overline{w}_F \ge w_F^*$ since we drop the rank-one restriction on $\boldsymbol{X}$ of MISDP formulation (40). On the other hand, for the case of $s = 2$, theorem 1.1 in [37] shows that for any feasible solution $(w, \boldsymbol{X}, \boldsymbol{z})$, there exists a rank-one semi-definite matrix $\widehat{\boldsymbol{X}}$ such that the new solution $(w, \widehat{\boldsymbol{X}}, \boldsymbol{z})$ is also feasible and achieves the same objective value. Thus, we must have $\overline{w}_F = w_F^*$;

(ii) Suppose $(w, \boldsymbol{X}, \boldsymbol{z})$ denotes an optimal solution of MISDP (40). Let $S = \{i \in [n] : z_i = 1\}$. Then according to theorem 1.7 in [37], there exists a semi-definite matrix $\widehat{\boldsymbol{X}}$ of the rank at most $1 + \lfloor \sqrt{2s + 9/4} - 3/2 \rfloor$ such that the new solution $(w, \widehat{\boldsymbol{X}}, \boldsymbol{z})$ is also optimal.

□

Proposition 8 shows that two-group SFPCA (39) admits an MISDP representation, while MISDP formulation (40) provides a low-rank solution in general for SFPCA when $s > 2$. It is worthy of mentioning that the results in Proposition 8 work for any convex fairness measure.

9. Conclusion In practice, to tune the parameter k via cross-validation, our developed greedy and local search algorithms can be quickly warm started from solution procedure in the previous iterations. We anticipate that the theoretical optimality gaps of three exact formulations for SPCA and R1-SSVD are not tight and can be further strengthened. The analysis of the optimality gap of sparse fair PCA requires new techniques, which can be an exciting research direction. Also, it might be desirable to study robust sparse PCA when the datasets are noisy or contain outliers.

References

[1] Amini AA, Wainwright MJ (2008) High-dimensional analysis of semidefinite relaxations for sparse principal components. *2008 IEEE International Symposium on Information Theory*, 2454–2458 (IEEE).

[2] Balas E (1975) Disjunctive programming: cutting planes from logical conditions. *Nonlinear Programming 2*, 279–312 (Elsevier).

[3] Ben-Tal A, Nemirovski A (2001) *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2 (Siam).

[4] Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* 4(1):238–252, ISSN 0029-599X, URL `http://dx.doi.org/10.1007/BF01386316`.

[5] Berk L, Bertsimas D (2019) Certifiably optimal sparse principal component analysis. *Mathematical Programming Computation* 11(3):381–420.

[6] Breakey WR, Goodell H, Lorenz PC, McHugh PR (1974) Hallucinogenic drugs as precipitants of schizophrenia. *Psychological Medicine* 4(3):255–261.

[7] Carrizosa E, Guerrero V (2014) rs-sparse principal component analysis: A mixed integer nonlinear programming approach with vns. *Computers & operations research* 52:349–354.

[8] Chaib S, Gu Y, Yao H (2015) An informative feature selection method based on sparse pca for vhr scene classification. *IEEE Geoscience and Remote Sensing Letters* 13(2):147–151.

[9] Chan SO, Papailliopoulos D, Rubinstein A (2016) On the approximability of sparse pca. *Conference on Learning Theory*, 623–646.

[10] Coope I (1994) On matrix trace inequalities and related topics for products of hermitian matrices. *Journal of mathematical analysis and applications* 188(3):999–1001.

[11] Coughlin LN, Tegge AN, Sheffer CE, Bickel WK (2020) A machine-learning approach to predicting smoking cessation treatment outcomes. *Nicotine and Tobacco Research* 22(3):415–422.

[12] d'Aspremont A, Bach F, Ghaoui LE (2012) Approximation bounds for sparse principal component analysis. *arXiv preprint arXiv:1205.0121* .

[13] d'Aspremont A, Ghaoui LE, Jordan MI, Lanckriet GR (2005) A direct formulation for sparse pca using semidefinite programming. *Advances in neural information processing systems*, 41–48.

[14] De Barona MS, Simpson DD (1984) Inhalant users in drug abuse prevention programs. *The American journal of drug and alcohol abuse* 10(4):503–518.

[15] Dey SS, Mazumder R, Wang G (2018) A convex integer programming approach for optimal sparse pca. *arXiv preprint arXiv:1810.09062* .

[16] d'Aspremont A, Bach F, Ghaoui LE (2008) Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research* 9(Jul):1269–1294.

[17] Gally T, Pfetsch ME (2016) Computing restricted isometry constants via mixed-integer semidefinite programming. *preprint, submitted* .

[18] Geoffrion AM (1972) Generalized benders decomposition. *Journal of optimization theory and applications* 10(4):237–260.

[19] He Y, Monteiro RD, Park H (2011) An algorithm for sparse pca based on a new sparsity control criterion. *Proceedings of the 2011 SIAM International Conference on Data Mining*, 771–782 (SIAM).

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

[21] Jiang R, Fei H, Huan J (2012) A family of joint sparse pca algorithms for anomaly localization in network data streams. *IEEE Transactions on Knowledge and Data Engineering* 25(11):2421–2433.

[22] Journée M, Nesterov Y, Richtárik P, Sepulchre R (2010) Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research* 11(2).

[23] Lee M, Shen H, Huang JZ, Marron J (2010) Biclustering via sparse singular value decomposition. *Biometrics* 66(4):1087–1095.

[24] Li Y, Xie W (2020) Best principal submatrix selection for the maximum entropy sampling problem: Scalable algorithms and performance guarantees. *arXiv preprint arXiv:2001.08537* .

[25] Luss R, d'Aspremont A (2010) Clustering and feature selection using sparse principal component analysis. *Optimization and Engineering* 11(1):145–157.

[26] Madan V, Singh M, Tantipongpipat U, Xie W (2019) Combinatorial algorithms for optimal design. *Conference on Learning Theory*, 2210–2258.

[27] Magdon-Ismail M (2017) Np-hardness and inapproximability of sparse pca. *Information Processing Letters* 126:35–38.

[28] Min W, Liu J, Zhang S (2016) L0-norm sparse graph-regularized svd for biclustering. *arXiv preprint arXiv:1603.06035* .

[29] Moghaddam B, Weiss Y, Avidan S (2006) Spectral bounds for sparse pca: Exact and greedy algorithms. *Advances in neural information processing systems*, 915–922.

[30] Naikal N, Yang AY, Sastry SS (2011) Informative feature selection for object recognition via sparse pca. *2011 International Conference on Computer Vision*, 818–825 (IEEE).

[31] Nedić A, Ozdaglar A (2009) Subgradient methods for saddle-point problems. *Journal of optimization theory and applications* 142(1):205–228.

[32] Ompad DC, Ikeda RM, Shah N, Fuller CM, Bailey S, Morse E, Kerndt P, Maslow C, Wu Y, Vlahov D, et al. (2005) Childhood sexual abuse and age at initiation of injection drug use. *American journal of public health* 95(4):703–709.

[33] Overdose O (2018) Understanding the epidemic. *Atlanta, Centers for Disease Control and Prevention* .

[34] Samadi S, Tantipongpipat U, Morgenstern JH, Singh M, Vempala S (2018) The price of fair pca: One extra dimension. *Advances in Neural Information Processing Systems*, 10976–10987.

[35] Semlyen A, Angelidis G (1995) Efficient calculation of critical eigenvalue clusters in the small signal stability analysis of large power systems .

[36] Sill M, Kaiser S, Benner A, Kopp-Schneider A (2011) Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics* 27(15):2089–2097.

[37] Tantipongpipat U, Samadi S, Singh M, Morgenstern JH, Vempala S (2019) Multi-criteria dimensionality reduction with applications to fairness. *Advances in Neural Information Processing Systems*, 15135–15145.

[38] Thomas DL, Vlahov D, Solomon L, Cohn S, Taylor E, Garfein R, Nelson KE (1995) Correlates of hepatitis c virus infections among injection drug users. *Medicine* 74(4):212–220.

[39] Volkow ND, Fowler JS, Wang GJ, Swanson JM, Telang F (2007) Dopamine in drug abuse and addiction: results of imaging studies and treatment implications. *Archives of neurology* 64(11):1575–1579.

[40] Zhang Y, d'Aspremont A, El Ghaoui L (2012) Sparse pca: Convex relaxations, algorithms and applications. *Handbook on Semidefinite, Conic and Polynomial Optimization*, 915–940 (Springer).

[41] Zhang Y, Ghaoui LE (2011) Large-scale sparse principal component analysis with application to text data. *Advances in Neural Information Processing Systems*, 532–539.

Appendix A. Proofs

A.1 Proof of Lemma 1

Lemma 1 *For a symmetric matrix $\boldsymbol{A} \in \mathcal{S}^n$ and a size-k set $S \subseteq [n]$, the followings must hold:*

(i) $\max_{\boldsymbol{x}\in\mathbb{R}^n} \left\{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} : ||\boldsymbol{x}||_2 = 1, x_i = 0, \forall i \notin S\right\} = \lambda_{\max}(\boldsymbol{A}_{S,S})$,

(ii) $\max_{\boldsymbol{X}\in\mathcal{S}_+^k} \{\mathrm{tr}(\boldsymbol{A}_{S,S}\boldsymbol{X}) : \mathrm{tr}(\boldsymbol{X}) = 1\} = \lambda_{\max}(\boldsymbol{A}_{S,S})$, *and*

(iii) If matrix $\boldsymbol{A}$ is positive semi-definite, then $\lambda_{\max}(\boldsymbol{A}_{S,S}) = \lambda_{\max}(\sum_{i\in S} \boldsymbol{c}_i\boldsymbol{c}_i^\top)$, where $\boldsymbol{A} = \boldsymbol{C}^\top\boldsymbol{C}$, $\boldsymbol{C} \in \mathbb{R}^{d\times n}$ denotes the Cholesky factorization matrix of $\boldsymbol{A}$, d is the rank of $\boldsymbol{A}$, and $\boldsymbol{c}_i \in \mathbb{R}^d$ denotes i-th column vector of $\boldsymbol{C}$ for each $i \in [n]$.

Proof. **Part (i)** Given a size-k set $S \subseteq [n]$, the maximization problem

$$\max_{\boldsymbol{x}\in\mathbb{R}^n} \left\{\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} : ||\boldsymbol{x}||_2 = 1, x_i = 0, \forall i \notin S\right\}$$

reduces to

$$\max_{\boldsymbol{x}\in\mathbb{R}^k} \left\{\boldsymbol{x}^\top \boldsymbol{A}_{S,S}\boldsymbol{x} : ||\boldsymbol{x}||_2 = 1\right\},$$

which is exactly the definition of the largest eigenvalue of principal submatrix $\boldsymbol{A}_{S,S}$.

Part (ii) According to Part (i), it is sufficient to show that $v^* = \widehat{v}$, where $v^*, \widehat{v}$ are defined as

$$v^* := \max_{\boldsymbol{X}\in\mathcal{S}_+^k} \{\mathrm{tr}(\boldsymbol{A}_{S,S}\boldsymbol{X}) : \mathrm{tr}(\boldsymbol{X}) = 1\}, \tag{41}$$

$$\widehat{v} := \max_{\boldsymbol{x}\in\mathbb{R}^k} \left\{\boldsymbol{x}^\top \boldsymbol{A}_{S,S}\boldsymbol{x} : ||\boldsymbol{x}||_2 = 1\right\}. \tag{42}$$

First, we must have $v^* \geq \widehat{v}$. Indeed, for any feasible $\boldsymbol{x} \in \mathbb{R}^k$ to problem (42) such that $||\boldsymbol{x}||_2 = 1$, we can construct a positive semi-finite matrix by $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$, which is feasible to problem (41) and yields the same objective value.

Second, to prove $\widehat{v} \geq v^*$, we let $\boldsymbol{X}^* \in \mathcal{S}_+^k$ denote an optimal solution to problem (41) and $\boldsymbol{X}^* = \sum_{i\in[k]} \lambda_i \boldsymbol{q}_i\boldsymbol{q}_i^\top$ denote its spectral decomposition. Since $\mathrm{tr}(\boldsymbol{X}^*) = 1$ and $\boldsymbol{X}^* \in \mathcal{S}_+^k$, the eigenvalues must satisfy $\sum_{i\in[k]} \lambda_i = 1$ and $\lambda_i \geq 0$ for each $i \in [k]$. Thus, the optimal value v^* of problem (41) is equal to

$$v^* = \mathrm{tr}(\boldsymbol{A}_{S,S}\boldsymbol{X}^*) = \sum_{i\in[k]} \lambda_i \boldsymbol{q}_i^\top \boldsymbol{A}_{S,S}\boldsymbol{q}_i \leq \max_{i\in[k]} \boldsymbol{q}_i^\top \boldsymbol{A}_{S,S}\boldsymbol{q}_i \leq \widehat{v},$$

where the inequality is due to $\sum_{i\in[k]} \lambda_i = 1$ and $\lambda_i \geq 0$ for each $i \in [k]$.

Part (iii) For a positive semi-definite matrix $\boldsymbol{A}$, let $\boldsymbol{A} = \boldsymbol{C}^\top\boldsymbol{C}$ denote the Cholesky factorization of $\boldsymbol{A}$ and $\boldsymbol{C} \in \mathbb{R}^{d\times n}$, thus we have

$$\lambda_{\max}(\boldsymbol{A}_{S,S}) = \lambda_{\max}(\boldsymbol{C}_S^\top\boldsymbol{C}_S) = \lambda_{\max}(\boldsymbol{C}_S\boldsymbol{C}_S^\top),$$

where the second equality is because for any matrix, its largest singular value is equal to that of its transpose. □

A.2 Proof of Proposition 1

Proposition 1 *For the function $H_1(\boldsymbol{z})$ defined in (9), we have*

(i) For any $\boldsymbol{z}\in\overline{Z}$, function $H_1(\boldsymbol{z})$ is equivalent to

$$H_1(\boldsymbol{z})=\min_{\boldsymbol{\mu},\boldsymbol{Q}_1,\cdots,\boldsymbol{Q}_n\in\mathcal{S}_+^d}\left\{\lambda_{\max}\Big(\sum_{i\in[n]}\boldsymbol{Q}_i\Big)+\sum_{i\in[n]}\mu_i z_i:\boldsymbol{c}_i\boldsymbol{c}_i^\top\preceq\boldsymbol{Q}_i+\mu_i\boldsymbol{I}_d,0\le\mu_i\le\|\boldsymbol{c}_i\|_2^2,\forall i\in[n]\right\},\tag{10}$$

which is concave in $\boldsymbol{z}$.

(ii) For any binary $\boldsymbol{z}\in Z$, an optimal solution to problem (10) is $\mu_i^*=0$ if $z_i=1$ and $\|\boldsymbol{c}_i\|_2^2$, otherwise, and $\boldsymbol{Q}_i^*:=(1-\mu_i^*/\|\boldsymbol{c}_i\|_2^2)\boldsymbol{c}_i\boldsymbol{c}_i^\top$ for each $i\in[n]$.

Proof. **Part (i).** We split the proof of strong duality into two cases depending on whether $\boldsymbol{z}$ is a relative interior point of set $\overline{Z}$ or not.

Case a. We will first prove the result by assuming that $\boldsymbol{z}$ is in the relative interior of set $\overline{Z}$, i.e., $0<z_i<1$ for each $i\in[n]$. For the inner maximization problem in (9), we dualize the constraint $\boldsymbol{X}\succeq\boldsymbol{W}_i,\mathrm{tr}(\boldsymbol{W}_i)=z_i$ with Lagrangian multiplier $\boldsymbol{Q}_i\in\mathcal{S}_+^d$ and μ_i for each $i\in[n]$. Note that the constraints $\boldsymbol{X}\succeq\boldsymbol{W}_i,\mathrm{tr}(\boldsymbol{W}_i)=z_i$ for each $i\in[n]$ and $\boldsymbol{X},\boldsymbol{W}_1,\cdots,\boldsymbol{W}_n\in\mathcal{S}_+^d$ can be always strictly satisfied since $0<z_i<1$. Thus, according to the strong duality of general conic program (see, e.g., Theorem 1.4.4 in [3]), function $H_1(\boldsymbol{z})$ can be rewrite as

$$\min_{\boldsymbol{\mu},\boldsymbol{Q}_1,\cdots,\boldsymbol{Q}_n\in\mathcal{S}_+^d}\max_{\boldsymbol{X},\boldsymbol{W}_1,\cdots,\boldsymbol{W}_n\in\mathcal{S}_+^d}\left\{\sum_{i\in[n]}\boldsymbol{c}_i^\top\boldsymbol{W}_i\boldsymbol{c}_i+\sum_{i\in[n]}\mathrm{tr}\left(\boldsymbol{Q}_i(\boldsymbol{X}-\boldsymbol{W}_i)\right)+\sum_{i\in[n]}\mu_i\left(z_i-\mathrm{tr}(\boldsymbol{W}_i)\right):\mathrm{tr}(\boldsymbol{X})=1\right\}.\tag{43}$$

Then the inner maximization problem (43) over $\boldsymbol{W}_i$ for each $i\in[n]$ and $\boldsymbol{X}$ yields

$$\max_{\boldsymbol{W}_i\in\mathcal{S}_+^d}\mathrm{tr}\left((\boldsymbol{c}_i\boldsymbol{c}_i^\top-\boldsymbol{Q}_i-\mu_i\boldsymbol{I}_d)\boldsymbol{W}_i\right)=\begin{cases}0, & \boldsymbol{c}_i\boldsymbol{c}_i^\top\preceq\boldsymbol{Q}_i+\mu_i\boldsymbol{I}_d,\\ \infty, & \text{otherwise.}\end{cases}$$
$$\max_{\boldsymbol{X}\in\mathcal{S}_+^d}\left\{\mathrm{tr}\left(\Big(\sum_{i\in[n]}\boldsymbol{Q}_i\Big)\boldsymbol{X}\right):\mathrm{tr}(\boldsymbol{X})=1\right\}=\lambda_{\max}\Big(\sum_{i\in[n]}\boldsymbol{Q}_i\Big),$$

where the second identity is due to Part(ii) of Lemma 1.

Thus, problem (43) can be simplified as

$$H_1(\boldsymbol{z})=\min_{\boldsymbol{\mu},\boldsymbol{Q}_1,\cdots,\boldsymbol{Q}_n\in\mathcal{S}_+^d}\left\{\lambda_{\max}\Big(\sum_{i\in[n]}\boldsymbol{Q}_i\Big)+\sum_{i\in[n]}\mu_i z_i:\boldsymbol{c}_i\boldsymbol{c}_i^\top\preceq\boldsymbol{Q}_i+\mu_i\boldsymbol{I}_d,\forall i\in[n]\right\}.\tag{44}$$

We show that for the minimization problem (44), any optimal solution $(\boldsymbol{\mu},\boldsymbol{Q}_1,\cdots,\boldsymbol{Q}_n)$ must satisfy $0\le\mu_i\le\|\boldsymbol{c}_i\|_2^2$ for each $i\in[n]$. We prove it by contradiction. Suppose that there exits an optimal solution $(\boldsymbol{\mu},\boldsymbol{Q}_1,\cdots,\boldsymbol{Q}_n)$ to the problem (44) such that $\mu_j<0$ for

some $j \in [n]$. Then, we can construct a new feasible solution $(\overline{\boldsymbol{\mu}}, \overline{\boldsymbol{Q}}_1, \cdots, \overline{\boldsymbol{Q}}_n)$, which is exactly equal to $(\boldsymbol{\mu}, \boldsymbol{Q}_1, \cdots, \boldsymbol{Q}_n)$ except

$$\overline{\mu}_j = 0, \overline{\boldsymbol{Q}}_j = \boldsymbol{Q}_j + \mu_j \boldsymbol{I}_d.$$

The new solution yields the objective value

$$H_1(\boldsymbol{z}) + \mu_j - \mu_j z_j = H_1(\boldsymbol{z}) + \mu_j(1 - z_j) < H_1(\boldsymbol{z}),$$

which is a contradiction to the optimality of $(\boldsymbol{\mu}, \boldsymbol{Q}_1, \cdots, \boldsymbol{Q}_n)$. Similarly, suppose that there exits an optimal solution $(\boldsymbol{\mu}, \boldsymbol{Q}_1, \cdots, \boldsymbol{Q}_n)$ to the problem (44) such that $\mu_j > \|\boldsymbol{c}_i\|_2^2$ for some $j \in [n]$. Similarly, we can arrive at a contradiction by defining a new feasible solution $(\overline{\boldsymbol{\mu}}, \overline{\boldsymbol{Q}}_1, \cdots, \overline{\boldsymbol{Q}}_n)$, which is exactly equal to $(\boldsymbol{\mu}, \boldsymbol{Q}_1, \cdots, \boldsymbol{Q}_n)$ except $\overline{\mu}_j = \|\boldsymbol{c}_i\|_2^2$.

Therefore, (44) can be reduced to (10).

Case b. Now we consider the case that $\boldsymbol{z}$ is not in the relative interior of $\overline{Z}$ and define two sets $T_0 := \{i \in [n] : z_i = 0\}$ and $T_1 := \{i \in [n] : z_i = 1\}$. Thus, at least one of the two sets is not empty. In this case, we first observe that $H_1(\boldsymbol{z})$ in (9) is equivalent to

$$H_1(\boldsymbol{z}) := \max_{\boldsymbol{X}, \boldsymbol{W}_1, \cdots, \boldsymbol{W}_d \in \mathcal{S}_+^d} \Bigg\{ \sum_{i \in [n] \setminus (T_0 \cup T_1)} \boldsymbol{c}_i^\top \boldsymbol{W}_i \boldsymbol{c}_i + \sum_{i \in T_1} \boldsymbol{c}_i^\top \boldsymbol{X} \boldsymbol{c}_i : \operatorname{tr}(\boldsymbol{X}) = 1, \\ \boldsymbol{X} \succeq \boldsymbol{W}_i, \operatorname{tr}(\boldsymbol{W}_i) = z_i, \forall i \in [n] \setminus (T_0 \cup T_1) \Bigg\}. \tag{45}$$

Next, applying the same procedure as Case a., we have

$$H_1(\boldsymbol{z}) = \min_{\boldsymbol{\mu}, \{\boldsymbol{Q}_i\}_{i \in [n] \setminus (T_0 \cup T_1)} \subseteq \mathcal{S}_+^d} \Bigg\{ \lambda_{\max}\Bigg(\sum_{i \in [n] \setminus (T_0 \cup T_1)} \boldsymbol{Q}_i + \sum_{i \in T_1} \boldsymbol{c}_i \boldsymbol{c}_i^\top \Bigg) + \sum_{i \in [n] \setminus (T_0 \cup T_1)} \mu_i z_i : \\ \boldsymbol{c}_i \boldsymbol{c}_i^\top \preceq \boldsymbol{Q}_i + \mu_i \boldsymbol{I}_d, 0 \le \mu_i \le \|\boldsymbol{c}_i\|_2^2, \forall i \in [n] \setminus (T_0 \cup T_1) \Bigg\}. \tag{46}$$

To show the equivalence between (46) and (10), it remains to prove that

$$\widehat{H}_1(\boldsymbol{z}) = \min_{\boldsymbol{\mu}, \{\boldsymbol{Q}_i\}_{i \in [n]} \subseteq \mathcal{S}_+^d} \Bigg\{ \lambda_{\max}\Bigg(\sum_{i \in [n]} \boldsymbol{Q}_i \Bigg) + \sum_{i \in [n]} \mu_i z_i : \\ \boldsymbol{c}_i \boldsymbol{c}_i^\top \preceq \boldsymbol{Q}_i + \mu_i \boldsymbol{I}_d, 0 \le \mu_i \le \|\boldsymbol{c}_i\|_2^2, \forall i \in [n] \Bigg\}. \tag{47}$$

First, given any feasible solution $(\boldsymbol{\mu}, \{\boldsymbol{Q}_i\}_{i \in [n] \setminus (T_0 \cup T_1)})$ to the problem (46), let us augment it by setting $\boldsymbol{Q}_i = \boldsymbol{0}, \mu_i = \|\boldsymbol{c}_i\|_2^2$ for each $i \in T_0$ and $\boldsymbol{Q}_i = \boldsymbol{c}_i \boldsymbol{c}_i^\top, \mu_i = 0$ for each $i \in T_1$. Then $(\boldsymbol{\mu}, \{\boldsymbol{Q}_i\}_{i \in [n]})$ is feasible to the problem (47) with the same objective value. Thus, we have $\widehat{H}_1(\boldsymbol{z}) \le H_1(\boldsymbol{z})$.

On the other hand, given any feasible solution $(\boldsymbol{\mu}, \{\boldsymbol{Q}_i\}_{i \in [n]})$ to the problem (47), then $(\boldsymbol{\mu}, \{\boldsymbol{Q}_i\}_{i \in [n] \setminus (T_0 \cup T_1)})$ is feasible to the problem (46) a smaller objective value since $\boldsymbol{c}_i \boldsymbol{c}_i^\top \preceq \boldsymbol{Q}_i + \mu_i$ for each $i \in T_1$. Thus, we have $\widehat{H}_1(\boldsymbol{z}) \ge H_1(\boldsymbol{z})$. This completes the proof.

Part (ii). For any $z \in Z$, let set S denote its support. We then construct a pair of the primal and dual solutions to the maximization problem in (9) and its dual (10) as

$$\boldsymbol{X}^* = \boldsymbol{q}_1 \boldsymbol{q}_1^\top, \boldsymbol{W}_i^* = \boldsymbol{X}^*, \forall i \in S, \boldsymbol{W}_i^* = 0, \forall i \in [n] \setminus S,$$
$$\boldsymbol{Q}_i^* = \boldsymbol{c}_i \boldsymbol{c}_i^\top, \mu_i = 0, \forall i \in S, \boldsymbol{Q}_i^* = 0, \mu_i = ||\boldsymbol{c}_i||_2^2, \forall i \in [n] \setminus S,$$

where $\boldsymbol{q}_1$ denote the eigenvector for the largest eigenvalue of matrix $\sum_{i \in S} \boldsymbol{c}_i \boldsymbol{c}_i^\top$.

According to the results in Lemma 1, the above solutions return the same objective value for primal and dual problems, which is $\lambda_{\max}(\sum_{i \in S} \boldsymbol{c}_i \boldsymbol{c}_i^\top)$. This proves the optimality of the proposed dual solution. □

A.3 Proof of Proposition 2

Proposition 2 *The SPCA* (2) *admits the following MISDP formulation:*

$$\text{(SPCA)} \quad w^* := \max_{\boldsymbol{z} \in Z, \boldsymbol{X} \in \mathcal{S}_+^n} \left\{ \operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X}) = 1, X_{ii} \le z_i, \forall i \in [n] \right\}. \tag{14}$$

and its continuous relaxation value is equal to $\lambda_{\max}(\boldsymbol{A})$.

Proof.

(i) To show the equivalence of problem (14) and SPCA (2), we only need to show that for any feasible $\boldsymbol{z} \in Z$ with its support $S = \{i : z_i = 1\}$, we must have

$$\max_{\boldsymbol{X} \in \mathcal{S}_+^n} \left\{ \operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X}) = 1, X_{ii} \le z_i, \forall i \in [n] \right\} = \lambda_{\max}(\boldsymbol{A}_{SS}). \tag{48}$$

Indeed, since $\boldsymbol{X}$ is a positive semi-definite matrix, thus $X_{ii} = 0$ for each $i \in [n] \setminus S$ implies

$$X_{ij} = 0, \forall (i,j) \notin S \times S.$$

The left-hand side of (48) is equivalent to

$$\max_{\boldsymbol{X} \in \mathcal{S}_+^n} \left\{ \operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X}) = 1, X_{ii} \le z_i, \forall i \in [n] \right\} = \max_{\boldsymbol{X} \in \mathcal{S}_+^k} \left\{ \operatorname{tr}(\boldsymbol{A}_{S,S}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X}) = 1 \right\} = \lambda_{\max}(\boldsymbol{A}_{SS}),$$

where the second equality is due to Part (ii) in Lemma 1.

(ii) The continuous relaxation value of problem (14) is

$$\overline{w}_3 = \max_{\boldsymbol{z} \in \overline{Z}, \boldsymbol{X} \in \mathcal{S}_+^n} \left\{ \operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X}) = 1, X_{ii} \le z_i, \forall i \in [n] \right\}.$$

Since $\operatorname{tr}(\boldsymbol{X}) = 1$, thus the linking constraint $X_{ii} \le z_i$ is redundant for each $i \in [n]$. Hence,

$$\overline{w}_3 = \max_{\boldsymbol{X} \in \mathcal{S}_+^n} \left\{ \operatorname{tr}(\boldsymbol{A}\boldsymbol{X}) : \operatorname{tr}(\boldsymbol{X}) = 1 \right\} = \lambda_{\max}(\boldsymbol{A}),$$

where the equality is due to Part (ii) in Lemma 1. □

A.4 Proof of Lemma 2

Lemma 2 *The following two inequalities are valid to SPCA* (14)

(i) $\sum_{j\in[n]} X_{ij}^2 \le X_{ii} z_i$ *for all* $i \in [n]$*; and*

(ii) $\left(\sum_{j\in[n]} |X_{ij}|\right)^2 \le k X_{ii} z_i$ *for all* $i \in [n]$.

Proof. From the proof of Proposition 2, there must exists an optimal solution $(\boldsymbol{z}^*, \boldsymbol{X}^*)$ of SPCA (14) such that $\boldsymbol{X}^*$ must be rank-one. Thus, without loss of generality, for any feasible solution $(\boldsymbol{z}, \boldsymbol{X})$ of SPCA (14), we can assume that $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$, where $(\boldsymbol{x}, \boldsymbol{z})$ is also feasible to SPCA (2).

Next, we split the proof into two parts.

(i) Since $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$, thus

$$\sum_{j\in[n]} X_{ij}^2 = \sum_{j\in[n]} x_i^2 x_j^2 = x_i^2 \le z_i X_{ii}, \forall i \in [n],$$

where the last inequality follows from the facts that $X_{ii} = x_i^2 \le z_i$ and z_i is binary for each $i \in [n]$.

(ii) It is known (see, e.g., [15]) that $||\boldsymbol{x}||_1 \le \sqrt{k}$. Thus,

$$\sum_{j\in[n]} |X_{ij}| = \sum_{j\in[n]} |x_i||x_j| \le \sqrt{k}|x_i| \le \sqrt{k}\sqrt{X_{ii} z_i},$$

where the second inequality is due to the facts that $X_{ii} = x_i^2 \le z_i$ and z_i is binary for each $i \in [n]$. □

A.5 Proof of Theorem 6

Theorem 6 *Given a threshold* $\epsilon > 0$*, the following MILP is* $O(\epsilon)$*-approximate to SPCA* (2)*, i.e.,* $\epsilon \le \widehat{w}(\epsilon) - w^* \le \epsilon\sqrt{d}$

$$\begin{aligned}
\widehat{w}(\epsilon) := \max_{w, \boldsymbol{z}\in Z, \boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{x}, , \boldsymbol{\delta}, \boldsymbol{\mu}, \boldsymbol{\sigma}} \; & w \\
\text{s.t.} \quad & \boldsymbol{x} = \boldsymbol{\delta}_{i1} + \boldsymbol{\delta}_{i2}, ||\boldsymbol{\delta}_{i1}||_\infty \le z_i, ||\boldsymbol{\delta}_{i2}||_\infty \le 1 - z_i, \forall i \in [n], \\
& \boldsymbol{x} = \sum_{j\in[d]} \boldsymbol{\sigma}_j, ||\boldsymbol{\sigma}_j||_\infty \le y_j, \sigma_{jj} = y_j, \forall j \in [d], \sum_{j\in[d]} y_j = 1, \\
& \boldsymbol{x} = \boldsymbol{\mu}_{\ell 1} + \boldsymbol{\mu}_{\ell 2}, ||\boldsymbol{\mu}_{\ell 1}||_\infty \le \alpha_\ell, ||\boldsymbol{\mu}_{\ell 2}||_\infty \le 1 - \alpha_\ell, \forall \ell \in [m], \\
& w = w_U - (w_U - w_L)\left(\sum_{i\in[m]} 2^{-i}\alpha_i\right), \\
& \left|\left|\sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1} - w_U \boldsymbol{x} + (w_U - w_L)\sum_{\ell\in[m]} 2^{-\ell}\boldsymbol{\mu}_{\ell 1}\right|\right|_\infty \le \epsilon, \\
& \boldsymbol{\alpha} \in \{0,1\}^m, \boldsymbol{y} \in \{0,1\}^d,
\end{aligned} \tag{22}$$

where w_L, w_U *separately denote the lower and upper bounds of SPCA,* $m := \lceil \log_2((w_U - w_L)\epsilon^{-1}) \rceil$ *and the infinite norm inequality constraints can be easily linearized.*

Proof. Throughout the proof, we use indices $i \in [n]$, $j \in [d]$, and $\ell \in [m]$ to denote the elements of three different dimensional vectors, respectively. To construct the MILP by SPCA (21) and show the approximation accuracy, we split the proof into four steps.

Step 1. Linearize the bilinear terms $\{z_i \boldsymbol{x}\}_{i\in[n]}$ in (21). This can be done by introducing two copies $\boldsymbol{\delta}_{i1}, \boldsymbol{\delta}_{i2}$ of vector $\boldsymbol{x}$ for each $i \in [n]$ such that

$$\boldsymbol{x} = \boldsymbol{\delta}_{i1} + \boldsymbol{\delta}_{i2}, ||\boldsymbol{\delta}_{i1}||_\infty \le z_i, ||\boldsymbol{\delta}_{i2}||_\infty \le 1 - z_i, \forall i \in [n], \sum_{i\in[n]} z_i \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{x} = \sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1}.$$

Step 2. Linearize the nonconvex constraint $\|\boldsymbol{x}\|_\infty = 1$. We first observe that due to symmetry, $\|\boldsymbol{x}\|_\infty = 1$ can be equivalently written as a disjunction with d sets as below

$$\cup_{j\in[d]} \big\{\boldsymbol{x} \in \mathbb{R}^d : x_j = 1, \|\boldsymbol{x}\|_\infty \le 1\big\}.$$

Next, for each $j \in d$, we introduce a binary variable $y_j = 1$ indicating the j-th set is active and 0, otherwise, and then create a copy $\boldsymbol{\sigma}_j \in \mathbb{R}^d$ of variable $\boldsymbol{x}$ such that

$$\boldsymbol{x} = \sum_{j\in[d]} \boldsymbol{\sigma}_j, ||\boldsymbol{\sigma}_j||_\infty \le y_j, \sigma_{jj} = y_j, \forall j \in [d], \sum_{j\in[d]} y_j = 1, \boldsymbol{y} \in \{0,1\}^d.$$

Step 3. Approximate and linearize bilinear term $w\boldsymbol{x}$. We first approximate variable w using m binary variables $\boldsymbol{\alpha} \in \mathbb{R}^m$ with $m := \lceil \log_2((w_U - w_L)/\epsilon) \rceil$. Thus, we have

$$w \approx w_U - (w_U - w_L)\bigg(\sum_{\ell\in[m]} 2^{-\ell}\alpha_\ell\bigg)$$

with approximation accuracy at most $(w_U - w_L)/2^m \le \epsilon$. The bilinear term $w\boldsymbol{x}$ is now approximated by

$$w\boldsymbol{x} \approx w_U \boldsymbol{x} - (w_U - w_L)\bigg(\sum_{\ell\in[m]} 2^{-\ell}\alpha_\ell \boldsymbol{x}\bigg). \tag{49}$$

With binary variables $\boldsymbol{\alpha}$, the resulting bilinear terms $\{\alpha_\ell \boldsymbol{x}\}_{\ell\in[m]}$ can be further linearized following the same arguments as Step 2, i.e.,

$$\begin{aligned}
&\boldsymbol{x} = \boldsymbol{\mu}_{\ell 1} + \boldsymbol{\mu}_{\ell 2}, ||\boldsymbol{\mu}_{\ell 1}||_\infty \le \alpha_\ell, ||\boldsymbol{\mu}_{\ell 2}||_\infty \le 1 - \alpha_\ell, \forall \ell \in [m],\\
&w_U \boldsymbol{x} - (w_U - w_L)\bigg(\sum_{\ell\in[m]} 2^{-\ell}\alpha_\ell \boldsymbol{x}\bigg) = w_U \boldsymbol{x} - (w_U - w_L)\sum_{\ell\in[m]} 2^{-\ell}\boldsymbol{\mu}_{\ell 1}.
\end{aligned}$$

Step 4. Finally, following the approximation and linearization results in Step 3, the equality constraint $\sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{\sigma}_{i1} = w\boldsymbol{x}$ in (21) might not hold exactly. Thus we replace the equality by the following inequality

$$\bigg|\bigg| \sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1} - w_U \boldsymbol{x} + (w_U - w_L)\sum_{i\in[m]} 2^{-i}\boldsymbol{\mu}_{i1} \bigg|\bigg|_\infty$$

$$= \left|\left| \sum_{i\in[n]} \boldsymbol{c}_i\boldsymbol{c}_i^\top z_i\boldsymbol{x} - w_U\boldsymbol{x} + (w_U - w_L)\sum_{i\in[m]} 2^{-i}\alpha_i\boldsymbol{x} \right|\right|_\infty$$
$$= \left|\left| w\boldsymbol{x} - w_U\boldsymbol{x} + (w_U - w_L)\sum_{i\in[m]} 2^{-i}\alpha_i\boldsymbol{x} \right|\right|_\infty \le (w_U - w_L)/2^m \le \epsilon,$$

which holds for any feasible solution of formulation (21).

First, we have $\widehat{w}(\epsilon) \ge w^* - \epsilon$ since $w := w^* - \epsilon$ is feasible to the MILP (22).

Moreover, given an optimal solution $(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{z}}, \widehat{w}(\epsilon))$ to the MILP (22), we must have

$$\left|\left| \sum_{i\in[n]} \widehat{z}_i\boldsymbol{c}_i\boldsymbol{c}_i^\top \widehat{\boldsymbol{x}} - \widehat{w}(\epsilon)\widehat{\boldsymbol{x}} \right|\right|_\infty \le \epsilon$$
$$(\Rightarrow) \quad \min_{\boldsymbol{x}:\|\boldsymbol{x}\|_\infty=1} \left|\left| \sum_{i\in[n]} \widehat{z}_i\boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{x} - \widehat{w}(\epsilon)\boldsymbol{x} \right|\right|_\infty \le \epsilon$$
$$(\Rightarrow) \quad d^{-1/2} \min_{\boldsymbol{x}:\|\boldsymbol{x}\|_\infty=1} \left|\left| \sum_{i\in[n]} \widehat{z}_i\boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{x} - \widehat{w}(\epsilon)\boldsymbol{x} \right|\right|_2 \le \epsilon$$
$$(\Rightarrow) \quad d^{-1/2} \min_{\boldsymbol{x}:\|\boldsymbol{x}\|_2\ge1} \left|\left| \sum_{i\in[n]} \widehat{z}_i\boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{x} - \widehat{w}(\epsilon)\boldsymbol{x} \right|\right|_2 \le \epsilon$$
$$(\Leftrightarrow) \quad d^{-1/2} \min_{\boldsymbol{x}:\|\boldsymbol{x}\|_2=1} \left|\left| \sum_{i\in[n]} \widehat{z}_i\boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{x} - \widehat{w}(\epsilon)\boldsymbol{x} \right|\right|_2 \le \epsilon$$

where the first implication is due to $\|\widehat{\boldsymbol{x}}\|_\infty = 1$, the second one is due to $\|\boldsymbol{x}\|_\infty \ge d^{-1/2}\|\boldsymbol{x}\|_2$ since $\boldsymbol{x} \in \mathbb{R}^d$, the third one is because $\|\boldsymbol{x}\|_\infty = 1$ implies $\|\boldsymbol{x}\|_2 \ge 1$, and the equivalence is because of monotonicity and positive homogeneity of the objective function. According to the last inequality, there exists an eigenvalue w of matrix $\sum_{i\in[n]} \widehat{z}_i\boldsymbol{c}_i\boldsymbol{c}_i^\top$ such that $|\widehat{w}(\epsilon) - w| \le \epsilon\sqrt{d}$, which further implies that $\widehat{w}(\epsilon) - w^* \le \epsilon\sqrt{d}$ since $w \le w^*$. □

A.6 Proof of Theorem 7

Theorem 7 *Given a threshold $\epsilon > 0$, by enforcing the binary variables $\boldsymbol{z}$ to be continuous, let $\overline{w}_5(\epsilon)$ denote the optimal value of the relaxed MILP formulation* (22)*. Then we have*

$$\overline{w}_5(\epsilon) \le \min\left\{k(\sqrt{d}/2 + 1/2),\ n/k\sqrt{d} + (n-k)(\sqrt{d}/2 + 1/2)\right\}w^* + \epsilon\sqrt{d}.$$

Proof. From the proof of Theorem 6, we know that $\overline{w}_5(\epsilon) \le \overline{w}_5(0) + \epsilon\sqrt{d}$. Thus, it is sufficient to show that

$$\overline{w}_5(0) \le k(\sqrt{d}/2 + 1/2)w^*.$$

We observe that when $\epsilon = 0$, the resulting formulation by relaxing binary variables $\boldsymbol{z}$ to be continuous becomes:

$$\overline{w}_5(0) = \max_{\substack{w,\boldsymbol{z}\in\overline{Z},\boldsymbol{x},\\ \{\boldsymbol{\delta}_{i1}\}_{i\in[n]},\{\boldsymbol{\delta}_{i2}\}_{i\in[n]}}} \left\{ w : \sum_{i\in[n]} \boldsymbol{c}_i\boldsymbol{c}_i^\top\boldsymbol{\delta}_{i1} = w\boldsymbol{x}, \|\boldsymbol{x}\|_\infty = 1, \right.$$

$$\boldsymbol{x}=\boldsymbol{\delta}_{i1}+\boldsymbol{\delta}_{i2}, ||\boldsymbol{\delta}_{i1}||_\infty \le z_i, ||\boldsymbol{\delta}_{i2}||_\infty \le 1-z_i, \forall i\in[n]\Big\}, \tag{50}$$

Next, we split the proof into three steps.

Step 1. For any feasible solution to problem (50), we have

$$\begin{aligned} w &= \frac{||\sum_{i\in[n]} \boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1}||_\infty}{||\boldsymbol{x}||_\infty} = ||\sum_{i\in[n]} \boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1}||_\infty \le \sum_{i\in[n]} ||\boldsymbol{c}_i\boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1}||_\infty = \sum_{i\in[n]} ||\boldsymbol{c}_i||_\infty |\boldsymbol{c}_i^\top \boldsymbol{\delta}_{i1}| \\ &\le \sum_{i\in[n]} ||\boldsymbol{c}_i||_\infty ||\boldsymbol{c}_i||_1 ||\boldsymbol{\delta}_{i1}||_\infty \le \sum_{i\in[n]} ||\boldsymbol{c}_i||_\infty ||\boldsymbol{c}_i||_1 z_i \le k \max_{i\in[n]} ||\boldsymbol{c}_i||_\infty ||\boldsymbol{c}_i||_1, \end{aligned}$$

where the first inequality is due to triangle inequality, the second one is because of Holder's inequality, the third one is because $||\boldsymbol{\delta}_{i1}||_\infty \le z_i$, and the last one is due to $||\boldsymbol{c}_i||_\infty ||\boldsymbol{c}_i||_1 \le \max_{j\in[n]} ||\boldsymbol{c}_j||_\infty ||\boldsymbol{c}_j||_1$ for each $i\in[n]$ and $\sum_{i\in[n]} z_i = k$.

Step 2. Now it remains to show that for each $i\in[n]$

$$||\boldsymbol{c}_i||_\infty ||\boldsymbol{c}_i||_1 \le \frac{\sqrt{d}+1}{2} w^*.$$

Let ς be a permutation of index set $[d]$ such that $c_{i,\varsigma(1)}, \cdots, c_{i,\varsigma(d)}$ are sorted in an ascending order. Then we have

$$c_{i,\varsigma(1)}^2 + \frac{1}{d-1}\Big(\sum_{j\in[2,d]} |c_{i,\varsigma(j)}|\Big)^2 \le c_{i,\varsigma(1)}^2 + \cdots + c_{i,\varsigma(d)}^2 = ||\boldsymbol{c}_i||_2^2 \le w^*,$$

where the first inequality is from the arithmetic and quadratic mean inequality and the second inequality follows from $||\boldsymbol{c}_i||_2^2 = \lambda_{\max}(\boldsymbol{c}_i\boldsymbol{c}_i^\top) \le w^*$.

For ease of exposition, let us introduce $v_1 = |c_{i,\varsigma(1)}|$ and $v_2 = \sum_{j\in[2,d]} |c_{i,\varsigma(j)}|$. Next, let us consider an optimization problem

$$\nu = \max_{\boldsymbol{v}\in\mathbb{R}_+^2} \Big\{ v_1(v_1+v_2) : v_1^2 + 1/(d-1)v_2^2 \le w^* \Big\}, \tag{51}$$

whose optimal value clearly provides an upper bound of $||\boldsymbol{c}_i||_\infty ||\boldsymbol{c}_i||_1$.

To solve (51), we first rewrite v_1, v_2 as

$$v_1 = r\sin(\theta) r, v_2 = r\sqrt{d-1}\cos(\theta), \theta\in[0,\pi/2], r\le\sqrt{w^*}.$$

In this way, the objective function (51) is equal to

$$\begin{aligned} v_1(v_1+v_2) = v_1^2 + v_1v_2 &= r^2\sin^2(\theta) + r^2\sqrt{d-1}\sin(\theta)\cos(\theta) = r^2\frac{1-\cos(2\theta)}{2} + r^2\sqrt{d-1}\frac{\sin(2\theta)}{2} \\ &= \frac{r^2}{2} - \frac{r^2}{2}\cos(2\theta) + \frac{1}{2}r^2\sqrt{d-1}\sin(2\theta) \le \frac{1}{2}r^2 + \frac{\sqrt{d}}{2}r^2 \le \frac{\sqrt{d}+1}{2}w^*, \end{aligned}$$

where the first inequality is due to Cauchy-Schwartz inequality and the second one is because $r^2 \le w^*$. Thus, we must have

$$||\boldsymbol{c}_i||_\infty ||\boldsymbol{c}_i||_1 \le \frac{\sqrt{d}+1}{2} w^*.$$

This proves the first bound $k(\sqrt{d}/2+1/2)$ together with Step 1.

Step 3. We now prove the second bound. Plugging the equations $\boldsymbol{\delta}_{i1} = \boldsymbol{x} - \boldsymbol{\delta}_{i2}$ for all $i \in [n]$, we rewrite the continuous relaxation value as

$$w = \frac{||\sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top (\boldsymbol{x} - \boldsymbol{\delta}_{i2})||_\infty}{||\boldsymbol{x}||_\infty} \leq \frac{||\sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{x}||_\infty}{||\boldsymbol{x}||_\infty} + \frac{||\sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{\delta}_{i2}||_\infty}{||\boldsymbol{x}||_\infty}$$
$$\leq \frac{||\sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top \boldsymbol{x}||_\infty}{||\boldsymbol{x}||_\infty} + (n-k)\frac{\sqrt{d}+1}{2} w^* \leq \max_{i\in[d]} \sum_{j\in[d]} |\overline{C}_{ij}| + (n-k)\frac{\sqrt{d}+1}{2} w^*,$$

where $\overline{\boldsymbol{C}} := \boldsymbol{C}\boldsymbol{C}^\top = \sum_{i\in[n]} \boldsymbol{c}_i \boldsymbol{c}_i^\top$ and the first inequality is from the triangle inequality, the second one follows from the derivations in Steps 1 and 2, and the third one is due to $x_i \leq 1$ for each $i \in [d]$.

Next, the first term of the right-hand side above can be upper bounded by

$$\max_{i\in[d]} \sum_{j\in[d]} |\overline{C}_{ij}| = ||\overline{\boldsymbol{C}}||_1 \leq \sqrt{d}||\overline{\boldsymbol{C}}||_2 = \sqrt{d}\lambda_{\max}(\overline{\boldsymbol{C}}) \leq \frac{n}{k}\sqrt{d} w^*,$$

where the equations are from the definition of ℓ_1-norm and ℓ_2-norm of a matrix and the second inequality is due to $\lambda_{\max}(\overline{\boldsymbol{C}}) = \lambda_{\max}(\boldsymbol{A}) \leq n/k w^*$. □

A.7 Proof of Lemma 3

Lemma 3 *Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, consider its augmented counterpart $\overline{\boldsymbol{A}}$ defined in (28), two integers $k_1 \in [m]$ and $k_2 \in [n]$, and three subsets $S, S_1, S_2 \subseteq [m+n]$ such that $S \subseteq [m+n], |S| = k_1 + k_2$, $S_1 = S \cap [m], |S_1| = k_1$ and $S_2 = S \cap [m+1, m+n], |S_2| = k_2$. Then the following identities must hold:*

(i) The eigenvalues of the augmented submatrix $\overline{\boldsymbol{A}}_{S,S}$ are the singular values of submatrix $\boldsymbol{A}_{S_1,S_2}$ and their negations;

(ii) $\sigma_{\max}(\boldsymbol{A}_{S_1,S_2}) = \lambda_{\max}(\overline{\boldsymbol{A}}_{S,S}) = 1/2 \max_{\boldsymbol{x}\in\mathbb{R}^{k_1+k_2}} \{\boldsymbol{x}^\top \overline{\boldsymbol{A}} \boldsymbol{x} : ||\boldsymbol{x}_{1:k_1}||_2 = 1, ||\boldsymbol{x}_{k_1+1:k_1+k_2}||_2 = 1\} = 1/2 \max_{\boldsymbol{X}\in\mathcal{S}_+^{k_1+k_2}} \left\{\mathrm{tr}(\overline{\boldsymbol{A}}_{S,S}\boldsymbol{X}) : \sum_{j\in[k_1]} X_{jj} = 1, \sum_{i\in[k_1+1,k_1+k_2]} X_{ii} = 1\right\}$.

Proof. The proof includes two parts.

(i) By the definition of augmented matrix $\overline{\boldsymbol{A}}$ in (28), for its submatrix $\overline{\boldsymbol{A}}_{S,S}$, we observe that

$$\overline{\boldsymbol{A}}_{S,S} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A}_{S_1,S_2} \\ \boldsymbol{A}_{S_1,S_2}^\top & \boldsymbol{0} \end{bmatrix}.$$

Then the statement in Part (i) directly follows from the result in Ben-Tal and Nemirovski [3], which shows that the eigenvalues of an augmented symmetric matrix exactly are equal to the singular values and negative ones of the original matrix.

(ii) The first equality $\lambda_{\max}(\overline{\boldsymbol{A}}_{S,S}) = \sigma_{\max}(\boldsymbol{A}_{S_1,S_2})$ is obtained from Part (i).

For the largest singular value of $\boldsymbol{A}_{S_1,S_2}$, we have

$$\begin{aligned}\sigma_{\max}(\boldsymbol{A}_{S_1,S_2}) &= \max_{\boldsymbol{u}\in\mathbb{R}^{k_1},\boldsymbol{v}\in\mathbb{R}^{k_2}} \left\{\boldsymbol{u}^\top \boldsymbol{A}_{S_1,S_2}\boldsymbol{v} : ||\boldsymbol{u}||_2 = 1, ||\boldsymbol{u}||_2 = 1\right\} \\ &= \frac{1}{2}\max_{\boldsymbol{x}\in\mathbb{R}^{k_1+k_2}} \left\{\boldsymbol{x}^\top \overline{\boldsymbol{A}}_{S,S}\boldsymbol{x} : ||\boldsymbol{x}_{1:k_1}||_2 = 1, ||\boldsymbol{x}_{k_1+1:k_1+k_2}||_2 = 1\right\}, \end{aligned} \tag{52}$$

which proves the second equality of Part (ii).

As for the last equality of Part (ii), we let $\widehat{w}^*_{\rm SVD}$ denote the optimal value of the right-hand side SDP problem. Then we must have $\widehat{w}^*_{\rm SVD} \geq \sigma_{\max}(\boldsymbol{A}_{S_1,S_2})$ as the SDP problem is exactly a SDP relaxation of the maximization problem over $\boldsymbol{x}$ in (52) by relaxing the rank-one constraint. On the other hand, summing up two constraints in the SDP problem, we obtain an upper bound of $\widehat{w}^*_{\rm SVD}$, i.e.,

$$\widehat{w}^*_{\rm SVD} \leq \frac{1}{2}\max_{\boldsymbol{X}\in\mathcal{S}_+^{k_1+k_2}} \left\{\mathrm{tr}(\overline{\boldsymbol{A}}_{S,S}\boldsymbol{X}) : \mathrm{tr}(\boldsymbol{X}) = 2\right\} = \lambda_{\max}(\overline{\boldsymbol{A}}_{S,S}) = \sigma_{\max}(\boldsymbol{A}_{S_1,S_2}),$$

where the first equality is due to Part (ii) in Lemma 1. □

A.8 Proof of Theorem 11

Theorem 11 *The continuous relaxation value $\overline{w}_{\rm SVD1}$ of formulation* (34) *satisfies*

$$w^*_{\rm SVD} \leq \overline{w}_{\rm SVD1} \leq \sqrt{mnk_1^{-1}k_2^{-1}}\, w^*_{\rm SVD}.$$

Proof. For the matrix $\overline{\boldsymbol{A}}^{\#}$ defined in (32), using Part (i) in Lemma 3, we can derive that its largest eigenvalue is equal to $2\sigma_{\max}(\boldsymbol{A})$. Let $(\widehat{\boldsymbol{z}}, \widehat{\boldsymbol{X}}, \widehat{\boldsymbol{W}}_1, \cdots, \widehat{\boldsymbol{W}}_{m+n})$ denote an optimal solution to the continuous SDP relaxation of problem (34). We now have

$$2\sigma_{\max}(\boldsymbol{A}) = \lambda_{\max}\left(\overline{\boldsymbol{A}}^{\#}\right) = \max_{\boldsymbol{X}\succeq 0, \mathrm{tr}(\boldsymbol{X})=1} \left\{\sum_{i\in[m+n]} \boldsymbol{c}_i^\top \boldsymbol{X}\boldsymbol{c}_i\right\} \geq \sum_{i\in[m+n]} \boldsymbol{c}_i^\top \widehat{\boldsymbol{X}}\boldsymbol{c}_i \geq \sum_{i\in[m+n]} \boldsymbol{c}_i^\top \widehat{\boldsymbol{W}}_i\boldsymbol{c}_i,$$

where the last inequality is because $\widehat{\boldsymbol{X}} \succeq \widehat{\boldsymbol{W}}_i$ for each $i \in [m+n]$. Note that the right-hand side above is equal to $\overline{w}_{\rm SVD1} + \sigma_{\max}(\boldsymbol{A})$ and the inequalities above lead to

$$\overline{w}_{\rm SVD1} = \sum_{i\in[m+n]} \boldsymbol{c}_i^\top \widehat{\boldsymbol{W}}_i\boldsymbol{c}_i - \sigma_{\max}(\boldsymbol{A}) \leq 2\sigma_{\max}(\boldsymbol{A}) - \sigma_{\max}(\boldsymbol{A}) = \sigma_{\max}(\boldsymbol{A}).$$

Now it remains to show that

Claim 1 $\sigma_{\max}(\boldsymbol{A}) \leq \sqrt{mnk_1^{-1}k_2^{-1}}\, w^*_{\rm SVD}$.

Proof. Let $\boldsymbol{u}_1$, $\boldsymbol{v}_1$ denote the top right and left eigenvectors of $\boldsymbol{A}$, i.e., $\boldsymbol{u}_1^\top \boldsymbol{A}\boldsymbol{v}_1 = \sigma_{\max}(\boldsymbol{A}), \boldsymbol{A}\boldsymbol{v}_1 = \sigma_{\max}(\boldsymbol{A})\boldsymbol{v}_1, \boldsymbol{u}_1^\top \boldsymbol{A} = \sigma_{\max}(\boldsymbol{A})\boldsymbol{u}_1^\top$. We tailor $\boldsymbol{u}_1$, $\boldsymbol{v}_1$ to meet the feasibility of R1-SSVD (27) as below

$$\widehat{u}_{j1} = \begin{cases} u_{j1}, & \text{if } u_{j1} \text{ is one of the } k_1 \text{ largest entries of } \boldsymbol{u}_1 \\ 0, & \text{otherwise} \end{cases}, \forall j \in [n],$$
$$\widehat{v}_{j1} = \begin{cases} (\boldsymbol{A}^\top \widehat{\boldsymbol{u}})_j, & \text{if } |(\boldsymbol{A}^\top \widehat{\boldsymbol{u}})_j| \text{ is one of the } k_2 \text{ largest entries of } |\boldsymbol{A}^\top \widehat{\boldsymbol{u}}| \\ 0, & \text{otherwise} \end{cases}, \forall j \in [m].$$

Let us normalize $\widehat{\boldsymbol{u}}_1 = \frac{\widehat{\boldsymbol{u}}_1}{||\widehat{\boldsymbol{u}}_1||_2}$ and $\widehat{\boldsymbol{v}}_1 = \frac{\widehat{\boldsymbol{v}}_1}{||\widehat{\boldsymbol{v}}_1||_2}$. Clearly, $(\widehat{\boldsymbol{u}}_1, \widehat{\boldsymbol{v}}_1)$ is feasible R1-SSVD (27). Then we have

$$\sqrt{\frac{k_1}{n}}\sigma_{\max}(\boldsymbol{A}) \le \sigma_{\max}(\boldsymbol{A})\widehat{\boldsymbol{u}}_1^\top \boldsymbol{u}_1 = \widehat{\boldsymbol{u}}_1^\top \boldsymbol{A}\boldsymbol{v}_1 \le \|\widehat{\boldsymbol{u}}_1^\top \boldsymbol{A}\|_2 \le \sqrt{\frac{m}{k_2}}\widehat{\boldsymbol{u}}_1^\top \boldsymbol{A}\widehat{\boldsymbol{v}}_1 \le \sqrt{\frac{m}{k_2}} w^*_{\text{SVD}},$$

where the first inequality is due to the definition of $\widehat{\boldsymbol{u}}_1$, the equality is because of the definition of $\boldsymbol{v}_1$, the second inequality is due to the Cauchy-Schwartz inequality, the third one is based on the choice of $\widehat{\boldsymbol{v}}_1$, and the last one is due to the feasibility of $(\widehat{\boldsymbol{u}}_1, \widehat{\boldsymbol{v}}_1)$. This completes the proof. ◇

□

A.9 Proof of Lemma 4

Lemma 4 *For R1-SSVD* (35)*, the following second-order conic inequalities are valid:*

(i) $\sum_{j\in[m]} X_{ij}^2 \le z_i X_{ii}, \sum_{j\in[m+1,m+n]} X_{ij}^2 \le z_i X_{ii}$ *for all* $i \in [m+n]$*; and*

(ii) $(\sum_{j\in[m]} |X_{ij}|)^2 \le k_1 X_{ii} z_i, (\sum_{j\in[m+1,m+n]} |X_{ij}|)^2 \le k_2 X_{ii} z_i$ *for all* $i \in [m+n]$.

Proof. According to Proposition 7, there must exist an optimal solution $(\boldsymbol{z}^*, \boldsymbol{X}^*)$ to MISDP (35) such that $\boldsymbol{X}^*$ is rank-one. Thus, without loss of generality, for any feasible solution $(\boldsymbol{z}, \boldsymbol{X})$ of SPCA (14), we can assume that $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix}^\top$, where vectors $(\boldsymbol{u}, \boldsymbol{v})$ thus satisfy

$$||\boldsymbol{u}||_2 = ||\boldsymbol{v}||_2 = 1, ||\boldsymbol{u}||_1 \le \sqrt{k_1}, ||\boldsymbol{v}||_1 \le \sqrt{k_2}.$$

Then the rest of the proof is almost identical to that of Lemma 2 and is thus omitted for brevity.

A.10 Proof of Theorem 15

Theorem 15 *For R1-SSVD* (27)*, the truncation algorithm yields an approximation ratio*

$$\max\left\{\sqrt{k_1^{-1}}, \sqrt{k_2^{-1}}, \sqrt{k_1 k_2 m^{-1} n^{-1}}\right\}.$$

In particular, the approximation ratio is $O(n^{-1/3})$ *when* $k_1 \approx k_2$ *and* $m \approx n$.

Proof. We derive the three approximation ratios of the truncation algorithm below.

(i) According to the truncation in the standard basis, the obtained vector $\widehat{\boldsymbol{u}}_i$ is feasible to the R1-SSVD problem for each $i \in [n]$ and is also optimal to the following problem

$$\widehat{\boldsymbol{u}}_i \in \arg \max_{||\boldsymbol{u}_i||_2=1, ||\boldsymbol{u}_i||_0=k_1} \left\{ \boldsymbol{u}_i^\top \boldsymbol{A} \boldsymbol{e}_i \right\}, \forall i \in [n].$$

Suppose the optimal solution of the R1-SSVD (27) to be $\boldsymbol{u}^*$ and $\boldsymbol{v}^*$, let S_1^*, S_2^* denote their supports, respectively. We then rewrite $\boldsymbol{v}^* = \sum_{i \in S_2^*} v_i^* \boldsymbol{e}_i$ and we have

$$w_{\rm SVD}^* = (\boldsymbol{u}^*)^\top \boldsymbol{A} \boldsymbol{v}^* = \sum_{i \in S_2^*} v_i^* (\boldsymbol{u}^*)^\top \boldsymbol{A} \boldsymbol{e}_i \le \sqrt{\sum_{i \in S_2^*} (v_i^*)^2} \sqrt{\sum_{i \in S_2^*} [(\boldsymbol{u}^*)^\top \boldsymbol{A} \boldsymbol{e}_i]^2} \le \sqrt{k_2} \max_{i \in [n]} \widehat{\boldsymbol{u}}_i^\top \boldsymbol{A} \boldsymbol{e}_i,$$

where the first inequality is due to Cauchy-Schwartz and the second one is because of maximality of $\max_{i \in [n]} \widehat{\boldsymbol{u}}_i^\top \boldsymbol{A} \boldsymbol{e}_i$.

Since $(1-(k_2-1)\epsilon)\boldsymbol{e}_i + \epsilon \sum_{j \in [k_2] \cup \{i\} \setminus \{i\}} \mathbf{e}_j$ with sufficiently small $\epsilon > 0$ is feasible to R1-SSVD (27), thus the right-hand side above is an lower bound of R1-SSVD according to the continuity by letting $\epsilon \to 0$. This prove the approximation ratio $\sqrt{k_2^{-1}}$.

Similarly, we can derive

$$w_{\rm SVD}^* = (\boldsymbol{u}^*)^\top \boldsymbol{A} \boldsymbol{v}^* \le \sqrt{k_1} \max_{j \in [m]} \boldsymbol{e}_j^\top \boldsymbol{A} \widehat{\boldsymbol{v}}_j,$$

which prove the approximation ratio $\sqrt{k_1^{-1}}$.

(ii) Following the proof of Claim 1, for the truncation in the eigen-space basis, we have

$$\sqrt{\frac{k_1}{n}} w_{\rm SVD}^* \le \sqrt{\frac{k_1}{n}} \sigma_{\max}(\boldsymbol{A}) \le \sigma_{\max}(\boldsymbol{A}) \widehat{\boldsymbol{u}}_1^\top \boldsymbol{u}_1 = \widehat{\boldsymbol{u}}_1^\top \boldsymbol{A} \boldsymbol{v}_1 \le \|\widehat{\boldsymbol{u}}_1^\top \boldsymbol{A}\|_2 \le \sqrt{\frac{m}{k_2}} \widehat{\boldsymbol{u}}_1^\top \boldsymbol{A} \widehat{\boldsymbol{v}}_1,$$

which proves the approximation ratio of $\sqrt{k_1 k_2 m^{-1} n^{-1}}$. □

A.11 Proof of Theorem 16

Theorem 16 *For the greedy Algorithm 3 and the local search Algorithm 4, we have (i) both algorithms achieve a $(\sqrt{k_1 k_2})^{-1}$-approximation ratio of R1-SSVD (37), and (ii) the ratio is tight.*

Proof. The proof is split into two parts.

(i) In R1-SSVD (37), according to the part (i) of the proof of Theorem 15, we have

$$w_{\rm SVD}^* \le \sqrt{k_2} \max_{j \in [n]} \widehat{\boldsymbol{u}}_j^\top \boldsymbol{A} \boldsymbol{e}_j \le \sqrt{k_1 k_2} \max_{i \in [m], j \in [n]} \boldsymbol{A}_{ij},$$

where vectors $\{\widehat{\boldsymbol{u}}_i\}_{i \in [n]} \subseteq \mathbb{R}^m$ are obtained by the normalized k_1-truncation in the standard basis of $\boldsymbol{A}$. Then, following the similar analyses of Theorem 8 and Theorem 9, the largest singular value from greedy Algorithm 3 and local search Algorithm 4 must be lower bounded by $\max_{i \in [m], j \in [n]} \boldsymbol{A}_{ij}$.

(ii) We next show an example in which the ratio $\sqrt{k_1^{-1}k_2^{-1}}$ can be achieved. Suppose that, without loss of generality, $k_1 \leq k_2$. Then, consider $m = 2k_2$, $n = 2k_2$, and matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ as

$$\boldsymbol{A} := \begin{bmatrix} \boldsymbol{I}_{k_2} & \boldsymbol{0}_{k_2\times k_2} \\ \boldsymbol{0}_{k_2\times k_2} & \boldsymbol{1}_{k_2\times k_2} \end{bmatrix}.$$

Above, the submatrix $\boldsymbol{A}_{[k_1],[k_2]}$ satisfies greedy and local optimality conditions with the objective value equal to 1, while the best size $k_1 \times k_2$ submatrix is $\boldsymbol{A}_{[k_2+1,k_2+k_1],[k_2+1,2k_2]}$ with the optimal value $\sqrt{k_1 k_2}$. □