Upper bounds for Model-Free Row-Sparse Principal Component Analysis

Guanyi Wang [1] **Santanu Dey** [1]

Abstract

Sparse principal component analysis (PCA) is a widely-used dimensionality reduction tool in statistics and machine learning. Most methods mentioned in literature are either heuristics for good primal feasible solutions under statistical assumptions or ADMM-type algorithms with stationary/critical points convergence property for the regularized reformulation of sparse PCA. However, none of these methods can efficiently verify the quality of the solutions via comparing current objective values with their dual bounds, especially in model-free case. We propose a new framework that finds upper (dual) bounds for the sparse PCA within polynomial time via solving a convex integer program (IP). We show that, in the worst-case, the dual bounds provided by the convex IP is within an affine function of the global optimal value. Moreover, in contrast to the semi-definition relaxation, this framework is much easier to scale for large instances. Numerical results on both artificial and real cases are reported to demonstrate the advantages of our method.

1. Introduction

Principal component analysis (PCA) is one of the most widely-used tool for dimensionality reduction and data visualization. Given a *sample matrix* $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M) \in \mathbb{R}^{d\times M}$ where each column denotes a d-dimensional zero-mean *sample*, the target is to find the top-r leading eigenvectors $\boldsymbol{V} := (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r) \in \mathbb{R}^{d\times r}$ (*principal components*),

$$\arg\max_{\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_r} \text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A} \boldsymbol{V}\right), \tag{PCA}$$

where $\boldsymbol{A} := \frac{1}{M}\boldsymbol{X}\boldsymbol{X}^\top$ is the *sample covariance matrix*, and $\boldsymbol{I}_r$ denotes the $r \times r$ identity matrix. However, a significant disadvantage with respect to interpretation of principal component analysis is that the principal component usually involves almost all components, especially in the high-dimensional setting, e.g. clinical analysis, biological gene analysis, computer version (**???**). Moreover, the principal component analysis is known to generate large generalization error, and therefore makes inaccurate prediction.

To enhance the interpretability, and reduce the generalization error, it is natural to consider the problem of sparse PCA via incorporating a sparsity constraint into the original PCA problem. There are many distinct descriptions of sparse PCA, mainly because the term "*sparsity*" can be defined in different ways based on the context. In this paper, we consider the *row-sparse PCA* problem (see for example (**?**)) defined as follows: Given a sample covariance matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, a *sparsity parameter* k ($\leq d$), the task is to find the top-r k-sparsity principal components $\boldsymbol{V} \in \mathbb{R}^{d\times r}$,

$$\arg\max_{\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_r,\ \|\boldsymbol{V}\|_0 \leq k} \text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A} \boldsymbol{V}\right), \tag{SPCA}$$

where the *row-sparsity constraint* $\|\boldsymbol{V}\|_0 \leq k$ denotes that there are at most k non-zero rows in matrix $\boldsymbol{V}$, i.e., the principal components share the *global support*. Let $\mathcal{F} := \{\boldsymbol{V} : \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_r,\ \|\boldsymbol{V}\|_0 \leq k\}$ denote the feasible region of SPCA and let $\text{opt}^{\mathcal{F}}(\boldsymbol{A})$ denote the optimal value of SPCA with sample covariance matrix $\boldsymbol{A}$.

1.1. "How good are primal feasible solutions?" why should we ask this question

Finding primal feasible solutions (with 'relatively' large objective value) for sparse PCA is well-studied in the field of statistics and optimization (see discussion of literature review below). Given a feasible solution, finding out its quality is a very important problem. We formalize the notion of good solution.

Definition 1.1 ($(1-\Delta)$-approximation primal feasible solution)**.** Let $\boldsymbol{V}_{\text{pri}} \in \mathbb{R}^{d\times r}$ be a primal feasible solution of SPCA with sample covariance matrix $\boldsymbol{A}$ and sparse parameter k. We say $\boldsymbol{V}_{\text{pri}}$ is a $(1-\Delta)$-*approximation primal feasible solution* if there is a $\Delta \in (0, 1)$ such that $\text{Tr}(\boldsymbol{V}_{\text{pri}}^\top \boldsymbol{A} \boldsymbol{V}_{\text{pri}}) \geq (1-\Delta)\text{opt}^{\mathcal{F}}(\boldsymbol{A})$.

*Equal contribution [1]H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, USA. Correspondence to: Guanyi Wang <gwang93@gatech.edu>, Santanu Dey <santanu.dey@isye.gatech.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

It is clear that if one can verify that Δ is close to 0 (without actually knowing the global optimal solution), the objective value corresponding to $\boldsymbol{V}$ is close to that of the global optimal. Therefore it is clear that there is value if attempting to obtaining the value of Δ in a model-free setting. However, usually we are working with a sample covariance matrix A which is only an estimate of a ground truth $\boldsymbol{\Sigma}$. Is it still true that finding the value of Δ is useful? The next proposition (will be formally stated and proved in Appendix) shows this to be true.

Proposition 1.1. Let samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ be i.i.d. generated from some underlying distribution with zero-mean and *true covariance matrix* $\boldsymbol{\Sigma}$. Let $\boldsymbol{A} := \frac{1}{M}\boldsymbol{X}\boldsymbol{X}$ be the sample covariance matrix defined as before, and $\boldsymbol{V}_{\text{app}}$ be a $(1-\Delta)$-approximation primal feasible solution of SPCA with respect to $\boldsymbol{A}$. If the number of samples M is sufficiently large, then:

$$\text{Tr}\left(\boldsymbol{V}_{\text{app}}^\top \boldsymbol{\Sigma} \boldsymbol{V}_{\text{app}}\right) \geq (1-\Delta)\text{opt}^{\mathcal{F}}(\boldsymbol{\Sigma}) - (2-\Delta)r\epsilon \quad (*)$$

holds with high probability, where ϵ is a constant that depends on the number of samples M.

Remark: Proposition 1.1 says that to verify the quality of a given primal feasible solution with respect to the true covariance matrix $\boldsymbol{\Sigma}$, it is sufficient to arrive at the value of Δ with respective to the sample covariance matrix $\boldsymbol{A}$. *Thus is sufficient to attempt to devise methods to find Δ in the model-free case, i.e. given a sample covariance matrix $\boldsymbol{A}$ and a primal solution $\boldsymbol{V}$, find Δ as in Definition 1.1 with out any assumption of underlying model.*

A natural idea to estimate Δ is by comparing $\text{opt}^{\mathcal{F}}(\boldsymbol{A})$ with $\text{Tr}(\boldsymbol{V}_{\text{app}}^\top \boldsymbol{A} \boldsymbol{V}_{\text{app}})$. Since we do not know the optimal solution, we do not know $\text{opt}^{\mathcal{F}}(\boldsymbol{A})$. Thus we have to find an upper bound to $\text{opt}^{\mathcal{F}}(\boldsymbol{A})$. Note that unlike PCA, there is no polynomial algorithm that achieves a constant multiplicative approximation ratio (**??**) (even when $r = 1$). Thus an efficient method to estimate the upper bound of SPCA is required.

Our Contributions: We present a convex relaxation of the feasible region $\mathcal{F}$ of SPCA. We use this convex relaxation to construct a second order cone integer programming relaxation of SPCA that can be solved in polynomial time. We prove the worst-case guarantees on upper bounds that can be obtained via solving this second order cone integer program. We propose a practical framework to solve the second order cone integer programming in practice. We also provide a new monotone search algorithm to find good primal feasible solutions. Numerical results are reported to illustrate the efficiency of our method (both in terms of finding good solutions and proving their high quality via dual bounds).

In rest of paper, we can use *primal bound*/*dual bound* to denote the lower bound/upper bound.

1.2. Literature Review

Existing approaches/results of solving/approximating the sparse PCA problem can be classified into the following categories:

In the first category, instead of dealing with the non-convex sparsity constraint directly, the papers (**????????**) incorporate additional regularizers to the objective function to enhance the sparsity of the solution. Similar to LASSO for sparse linear regression problem, these new formulations can be optimized efficiently via alternating-minimization type algorithms. However, the optimization problem presented in (**?**) is NP-hard to solve, and there is no convergence guarantee for the alternating-minimization method given in (**?**). The papers (**?**), (**?**), (**?**), (**?**), (**?**), (**?**) propose their own formulations for sparse PCA problem, and show that the alternating-minimization algorithm converges to stationary (critical) points. However, the solutions obtained using the above methods cannot guarantee the row-sparsity constraint $\|\boldsymbol{V}\|_0 \leq k$. Moreover, none of these methods are able to provide worst-case guarantees.

The second category of methods work with the sparsity constraint or its relaxation. The papers (**????**) directly incorporate the sparsity constraint (for $r = 1$ case) and then relax the resulting optimization problem into some convex optimization problems via semi-definite programming (SDP). However, SDPs are usually difficult to scale to large instance in practice. To be more scalable, (**??**) propose frameworks to find the dual bounds of sparse PCA problem using convex integer programming for the $r = 1$ case. A special case is when we have the low-rank sample covariance matrix $\boldsymbol{A}$. (**?**) proposes an exact algorithm to find the global optimal solution of SPCA with $r = 1$ and the total computation complexity of $O(d^{\text{rank}(\boldsymbol{A})+1}\log d)$. Later the paper (**?**) gives a combinatorial method for multi-component sparse PCA problem with *disjoint* supports. They show that their algorithm outputs a feasible solution within $(1-\epsilon)$-multiplicative approximation ratio in time polynomial in data dimension d and reciprocal of ϵ, but exponential in the rank of sample covariance matrix $\boldsymbol{A}$ and parameter r. (**?**) provides a general method for solving SPCA exactly with computational complexity polynomial in d, but exponential in r and $\text{rank}(\boldsymbol{A})$. The paper (**?**) clearly states that the results obtained are of theoretical nature for the low rank case, although these methods may not be practically implementable.

Some specialized iterative methods have been proposed to find a primal feasible leading sparse eigenvector in (**??????**), but the solutions of these methods are obtained from some deflation step similar to (**?**), thus these methods fail to deal with the row-sparsity condition.

Under the assumption of an underlying statistical model,

the paper (**?**) presents a family of estimators for SPCA with oracle property, using semidefinite relaxation of sparse PCA with decomposable non-convex penalty. The paper (**?**) analyzes a covariance thresholding algorithm (first proposed by (**?**)). They show that this algorithm correctly recovers the support with high probability for sparse parameter k in order $\sqrt{M}$ with M being the number of samples. This sample complexity, combining with the lower bounds results in (**??**), suggest that no polynomial time algorithm can do significantly better under their statistical assumptions. There are also a series of papers (**?????**) that provide the minimax rate of estimation for sparse PCA. However, all these papers require underlying statistical models, thus do not have worst-case guarantees in the model-free case.

1.3. Organization

The rest of the paper is organized as follows: The main theoretical results are presented in Section 2. In Section 2.1, we construct an SDP and SOCP representable convex relaxation of the feasible region. We also prove worst-case approximation ratio guarantee when optimizing over these convex feasible regions. In Section 2.2, we construct a SOCPIP relaxation for SPCA based on the previous subsection's results. We also provide an analysis for the worst-case guarantees on upper bounds when optimizing over SOCPIP. Next in sub-Section 2.3, we propose a monotone increasing search algorithm which is able to find a good primal feasible solution efficiently in practice. In sub-Section 2.4, we propose a practical framework for model-free SPCA to solve SOCPIP. Finally in Section 3, we compare the numerical results obtained from the SOCIP-impl framework against other methods to obtain dual bounds on two types of instances.

1.4. Notations

Let the bold upper case letters, for example, $\boldsymbol{A}, \boldsymbol{B}$ be matrices, and denote its (i,j)-th component as $[\boldsymbol{A}]_{ij}, [\boldsymbol{B}]_{ij}$. Let $\text{supp}(\boldsymbol{A})$ be the support of non-zero rows of matrix $\boldsymbol{A}$. Let the bold lower case letters, for example, $\boldsymbol{a}, \boldsymbol{b}$ be vectors, and denote its i-th component as $[\boldsymbol{a}]_i, [\boldsymbol{b}]_i$. Let the regular case letters, for example, I, J be the set of indices. For an integer k, let $[k] := \{1, \ldots, k\}$. Given any matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ and $I \subseteq [n], J \subseteq [m]$, let $[\boldsymbol{A}]_{I,J}$ (or $\boldsymbol{A}_{I,J}$ in short) be the sub-matrix of A with rows in I and columns in J. In order to simplify notation, let $[\boldsymbol{A}]_I$ be the sub-matrix of A when considering all rows in I, and let $[\boldsymbol{A}]_j$ be the jth row of matrix $\boldsymbol{A}$. Let the regular lower case letters, for example, α, β be the reals. Let $\oplus$ be the sign of direct plus, i.e., $\boldsymbol{A}, \boldsymbol{B}$, $\boldsymbol{A} \oplus \boldsymbol{B} := \text{diag}(\boldsymbol{A}, \boldsymbol{B})$.

2. Main Theoretical Results

We present our main theoretical results in this section.

2.1. Convex Relaxation for Feasible Region

Recall the feasible region of SPCA, denoted as $\mathcal{F}$ is:

$$\mathcal{F} := \left\{ \boldsymbol{V} \in \mathbb{R}^{d\times r} : \begin{array}{ll} \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_r & (1) \\ \|\boldsymbol{V}\|_0 \le k & (2) \end{array} \right\},$$

where the constraint (1) is the so-called *Stiefel manifold* (**?**) denoted as $\text{St}(d,r)$, and the constraint (2) is the row-sparsity constraint. For the constraint (1), it is well-known (**?**) that the convex hull of the Stiefel manifold $\text{conv}\left(\text{St}(d,r)\right)$ can be represented explicitly as $\text{conv}(\text{St}(d,r)) = \{\boldsymbol{V} : \boldsymbol{I}_r - \boldsymbol{V}^\top \boldsymbol{V} \succeq \boldsymbol{0}_r\}$ (SDP format) or $\text{conv}(\text{St}(d,r)) = \{\boldsymbol{V} : \|\boldsymbol{V}\|_{\text{op}} \le 1\}$ (operator norm format). For the constraint (2),

Proposition 2.1. If $\boldsymbol{V} \in \mathcal{F}$, then $\|[\boldsymbol{V}]_{[d],i}\|_1 \le \sqrt{k}$ holds for all $i \in [r]$.

The above proposition can be viewed as the ℓ_1-relaxation of the sparsity constraint for each column in $\boldsymbol{V}$. Moreover, the row-sparsity property can be futher captured by

Proposition 2.2. If $\boldsymbol{V} \in \mathcal{F}$, then $\sum_{j=1}^d \|[\boldsymbol{V}]_{j,[r]}\|_2 \le \sqrt{rk}$.

From Proposition 2.1 and 2.2, we obtain the following result.

Corollary 2.1. [SDP-relaxation] Let $\mathcal{F}$ be the feasible region of SPCA. We have $\text{conv}(\mathcal{F})$ is contained in the following convex set

$$\mathcal{C} := \left\{ \boldsymbol{V} : \begin{array}{l} \boldsymbol{I}_r - \boldsymbol{V}^\top \boldsymbol{V} \succeq \boldsymbol{0}_r, \\ \sum_{j=1}^d \|\boldsymbol{v}_i\|_1 \le \sqrt{k},\ \forall i \in [r] \\ \sum_{j=1}^d \|[\boldsymbol{V}]_{j,[r]}\|_2 \le \sqrt{rk},\ \forall j \in [d] \end{array} \right\}.$$

Since SDP-relaxations are usually difficult to solve, to be more scalable in practice, instead of using semi-definite constraint, we replace it with second-order cone constraints. In particular, we will replace the constraints defining the convex hull of the Stiefel manifold by a simple second-order-cone representable relaxation to obtain the following result.

Corollary 2.2. [SOCP-relaxation] Let $\mathcal{F}$ be the feasible region of SPCA. We have $\text{conv}(\mathcal{F})$ is contained in the following convex set

$$\mathcal{C}' := \left\{ \boldsymbol{V} : \begin{array}{l} \|[\boldsymbol{V}]_{[d],i}\|_2^2 \le 1,\ \forall i \in [r] \\ \|[\boldsymbol{V}]_{[d],i_1} \pm [\boldsymbol{V}]_{[d],i_2}\|_2^2 \le 2,\ \forall i_1 \ne i_2 \in [r] \\ \sum_{j=1}^d \|[\boldsymbol{V}]_{j,[r]}\|_2 \le \sqrt{rk} \\ \|[\boldsymbol{V}]_{j,[r]}\|_2 \in [0,1],\ \forall j \in [d] \end{array} \right\}.$$

Let $\text{opt}^{\mathcal{F}}, \text{opt}^{\mathcal{C}}, \text{opt}^{\mathcal{C}'}$ be the optimal values of the follow-

ing:

$$\begin{aligned} \text{opt}^{\mathcal{F}} &:= \max_{\boldsymbol{V}\in\mathcal{F}} \ \text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right), \\ \text{opt}^{\mathcal{C}} &:= \max_{\boldsymbol{V}\in\mathcal{C}} \ \text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right), && \text{(Relax)} \\ \text{opt}^{\mathcal{C}'} &:= \max_{\boldsymbol{V}\in\mathcal{C}'} \ \text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right). && \text{(SOCP-Relax)} \end{aligned}$$

Our first main result is that:

Theorem 1. $\text{opt}^{\mathcal{F}} \le \text{opt}^{\mathcal{C}'} \le (1+\sqrt{r})^2 \text{opt}^{\mathcal{F}}$.

Corollary 2.3. $\text{opt}^{\mathcal{F}} \le \text{opt}^{\mathcal{C}} \le (1+\sqrt{r})^2 \text{opt}^{\mathcal{F}}$.

Remark: *For $r = 1$ case, Theorem 1 and Corollary 2.3 provide constant multiplicative approximation ratios. Thus inapproximability results from (**??**) implies that solving Relax or SOCP-Relax to optimality is NP-hard.*

2.2. Upper (Dual) Bounds for SPCA

The challenge of solving SOCP-Relax is that the objective function is non-convex. Moreover, as the previous remark suggests, solving SOCP-Relax is NP-hard. Therefore we construct a further relaxation for SOCP-Relax. Since the only non-convex part is its objective function, i.e., maximizing a convex quadratic function, we proceed as follows:

Let $\boldsymbol{A} = \sum_{j=1}^{d} \lambda_j \boldsymbol{a}_j \boldsymbol{a}_j^\top$ be the eigenvalue decomposition of sample covariance matrix $\boldsymbol{A}$ with $\lambda_1 \ge \cdots \ge \lambda_d \ge 0$. The objective function then can be represented as a summation $\text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right) = \sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} (\boldsymbol{a}_j^\top \boldsymbol{v}_i)^2$ where $\boldsymbol{v}_i$ denotes the ith column of $\boldsymbol{V}$ such that $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r)$. Set auxiliary variables $g_{ji} = \boldsymbol{a}_j^\top \boldsymbol{v}_i$ for $(j,i) \in [r] \times [d]$. Let $\boldsymbol{a}_j \in \mathbb{R}^d$ satisfy $|[\boldsymbol{a}_j]_{j_1}| \ge \ldots \ge |[\boldsymbol{a}_j]_{j_k}| \ge \ldots \ge |[\boldsymbol{a}_j]_{j_d}|$, and let $\theta_j = \sqrt{[\boldsymbol{a}_j]_{j_1}^2 + \cdots + [\boldsymbol{a}_j]_{j_k}^2}$ be the square root of sum of top-k largest absolute entries. Since $\boldsymbol{v}_i$ is supposed to be k-sparse, it is easy to observe that g_{ji} is within the interval $[-\theta_j, \theta_j]$.

Piecewise Linear Approximation: To relax the non-convex part, we can upper approximate each quadratic term g_{ji}^2 by a piecewise linear function based on a new auxiliary variable ξ_{ji} via *special ordered sets type 2* (SOS-II, see Appendix for details) constraints (PLA) as follows,

$$\text{PLA} := \left\{ (g,\xi,\eta): \begin{array}{l} g_{ji} = \boldsymbol{a}_j^\top \boldsymbol{v}_i, \ (j,i) \in [d]\times[r] \\ g_{ji} = \sum_{\ell=-N}^{N} \gamma_{ji}^{\ell} \eta_{ji}^{\ell} \\ \xi_{ji} = \sum_{\ell=-N}^{N} \left(\gamma_{ji}^{\ell}\right)^2 \eta_{ji}^{\ell} \\ \left(\eta_{ji}^{\ell}\right)_{\ell=-N}^{N} \in \text{SOS-II} \end{array} \right\}$$

where for each $(j,i) \in [d]\times[r]$, $\left(\eta_{ji}^{\ell}\right)_{\ell=-N}^{N}$ is the corresponding set of SOS-II *integer* variables (see Appendix), and $\left(\gamma_{ji}^{\ell}\right)_{\ell=-N}^{N}$ is the corresponding set of splitting points that satisfy:

$$\underbrace{\gamma_{ji}^{-N}}_{=-\theta_j} \le \cdots \le \underbrace{\gamma_{ji}^{0}}_{=0} \le \cdots \le \underbrace{\gamma_{ji}^{N}}_{=\theta_j}$$

which split the region $[-\theta_j, \theta_j]$ into $2N$ intervals. See Figure 1 for an example. Thus the objective function can be

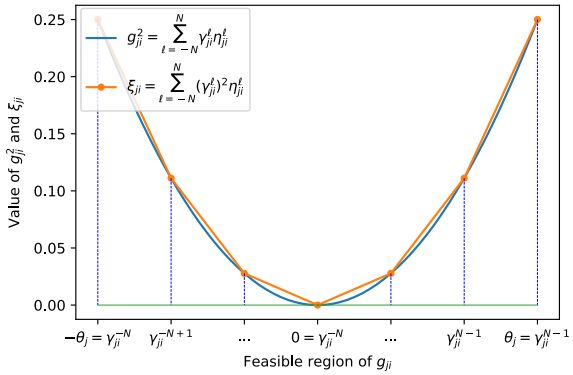

Figure 1. The quadratic term g_{ji}^2 is upper approximated by ξ_{ji} via SOS-II constraints for all $(j,i) \in [d] \times [r]$.

upper bounded by $\sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} \xi_{ji}$, and we can transfer this non-convex optimization problem into the following *second order cone convex integer program*,

$$\begin{aligned} \text{ub}^{\mathcal{C}'} := \max \quad & \textstyle\sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} \xi_{ji} \\ \text{s.t.} \quad & \boldsymbol{V} \in \mathcal{C}', \ (g,\xi,\eta) \in \text{PLA} \end{aligned} \qquad \text{(SOCIP)}$$

where $\mathcal{C}'$ is defined in Section 2.1 as a convex set such that $\text{Conv}(\mathcal{F}) \subseteq \mathcal{C}'$, and PLA is the set of constraints for piecewise-linear upper approximation. *We say this is a convex integer program since SOS-II constraints contain integer variables* (η_{ij}^{ℓ}). Therefore we arrive at the following results:

Proposition 2.3. The optimal value $\text{ub}^{\mathcal{C}'}$ of SOCIP is an upper bound of SPCA.

Proposition 2.4. The optimal value $\text{ub}^{\mathcal{C}'}$ of SOCIP can be upper bounded by $\text{ub}^{\mathcal{C}'} \le (1+\sqrt{r})^2 \text{opt}^{\mathcal{F}} + \sum_{j=1}^{d} \frac{r\lambda_j \theta_j^2}{4N^2}$, which is an affine function of $\text{opt}^{\mathcal{F}}$.

2.3. Lower (Primal) Bounds for SPCA

In this section, we will present a new algorithm that produces good solutions for SPCA. The quality of these solutions will be tested via comparison with the upper bound obtained by solving SOCIP. Moreover, in the next section, we show how to use the lower bound to reduce the running time of SOCIP.

Two-stage Idea: We are motivated by similar ideas for LASSO in sparse linear regression: Given a sample covariance matrix $\boldsymbol{A}$, let $\boldsymbol{A}^{1/2}$ be its positive semi-definite

square root such that $\boldsymbol{A} = \boldsymbol{A}^{1/2}\boldsymbol{A}^{1/2}$, the SPCA can be represented in the following fashion:

$$\begin{array}{ll} \min_{\boldsymbol{V}\in\mathbb{R}^{d\times r}} & \left\|\boldsymbol{A}^{1/2} - \boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{A}^{1/2}\right\|_F^2 \\ \text{s.t.} & \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}_r,\ \|\boldsymbol{V}\|_0 \le k \end{array} \quad \text{(SPCA-lasso)}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Let $S = \text{supp}(\boldsymbol{V})$ be the index set of the support of non-zero rows of principal components, and let $S^C = [d]\backslash S$ be the complement set of S in $[d]$. The SPCA-lasso can be reformulated into a *two-stage (inner & outer) optimization problem*:

$$\begin{array}{lll} \min_{S\subseteq[d],\ |S|\le k} & \min_{[\boldsymbol{V}]_S} & f(S,[\boldsymbol{V}]_S) \\ & \text{s.t.} & [\boldsymbol{V}]_S^\top[\boldsymbol{V}]_S = \boldsymbol{I}_r \end{array}$$

with $f(S,[\boldsymbol{V}]_S) := \|[\boldsymbol{A}^{1/2}]_S - [\boldsymbol{V}]_S[\boldsymbol{V}]_S^\top[\boldsymbol{A}^{1/2}]_S\|_F^2 + \|[\boldsymbol{A}^{1/2}]_{S^C}\|_F^2$. Given support S, there is a closed form solution of the inner optimization by eigenvalue decomposition of $[\boldsymbol{A}]_{S,S} = [\boldsymbol{A}^{1/2}]_S[\boldsymbol{A}^{1/2}]_S^\top$. Let $[\boldsymbol{A}]_{S,S} = \boldsymbol{U}_S\boldsymbol{\Lambda}_S\boldsymbol{U}_S^\top$ with eigenvalues in $\boldsymbol{\Lambda}_S$ ordered decreasingly. Set

$$[\boldsymbol{V}]_S = [\boldsymbol{U}_S]_{[k],[r]} \quad \text{(update-V)}$$

and note that this achieves the global minimal of the inner optimization. Thus the main challenge of solving SPCA (or SPCA-lasso equivalently) is to find a support S within $\binom{d}{k}$ possible support set.

Algorithm Intuition: *The intuition to find a relatively good primal solution is via a local search method which guarantees to reduce the objective value of SPCA-lasso in each epoch.*

Removing Candidate: Start with a support set S_0 (see Appendix for initialization). In t-th epoch, we have the support set $S_{t-1} = \{j_1,\dots,j_k\}$ and corresponding principal components $[\boldsymbol{V}]_{S_{t-1}}$ from $(t-1)$-th epoch. For each index $j_p \in S_{t-1}$ with $p = 1,\dots,k$, let the *reduced value* Δ_{j_p} be

$$\|[\boldsymbol{A}^{1/2}]_{j_p}\|_2^2 - \left\|\left[[\boldsymbol{A}^{1/2}]_{S_{t-1}} - [\boldsymbol{V}]_{S_{t-1}}[\boldsymbol{V}]_{S_{t-1}}^\top[\boldsymbol{A}^{1/2}]_{S_{t-1}}\right]_p\right\|_2^2,$$

which denotes how much the j_p-th row 'reduced'. To update the support set S_{t-1}, our method 1 determines the index $j^{\text{out}} := \arg\min_{j_p\in S_{t-1}}\Delta_{j_p}$. We let j^{out} to be the candidate index to be removed from S_{t-1}.

Entering Candidate: Similarly, for each index $j_q \in S_{t-1}^C$ with $q = 1,\dots,d-k$, let the *will-reduced value* Δ_{j_q} be

$$\|[\boldsymbol{A}^{1/2}]_{j_q}\|_2^2 - \left\|\left[[\boldsymbol{A}^{1/2}]_{S_{t-1}^{j_q}} - [\boldsymbol{V}]_{S_{t-1}}[\boldsymbol{V}]_{S_{t-1}}^\top[\boldsymbol{A}^{1/2}]_{S_{t-1}^{j_q}}\right]_q\right\|_2^2,$$

where $S_{t-1}^{j_q}$ is defined as $S_{t-1}^{j_q} := S_{t-1} - \{j^{\text{out}}\} + \{j_q\}$. Pick $j^{\text{in}} := \arg\max_{j_q\in S_{t-1}^C}\Delta_{j_q}$ such that the j^{in}-th row with the maximal reduction by entering S_{t-1}. Let j^{in} be the candidate of entering into S_{t-1}.

Updating Rule: We update the support set from S_{t-1} to S_t as follows

$$S_t = \begin{cases} S_{t-1} - \{j^{\text{out}}\} + \{j^{\text{in}}\} & \text{if } \Delta_{j^{\text{out}}} < \Delta_{j^{\text{in}}}, \\ S_{t-1} & \text{if } \Delta_{j^{\text{out}}} < \Delta_{j^{\text{in}}}, \end{cases}$$

and update $\boldsymbol{V}_{S_t}$ by update-V. Here is the pseudocode:

Algorithm 1 Local Search Method

Input: Covariance matrix $\boldsymbol{A}$, sparsity parameter k, number of eigenvectors r, number of maximum iterations T.
Output: A feasible solution $\boldsymbol{V}$ for SPCA.

t
while epoch $t = 1,\dots,T$ **do**
  For each $j \in S_{t-1}$, set the reduced value Δ_j.
  Set removing candidate $j^{\text{out}} := \arg\min_{j\in S_{t-1}}\Delta_j$.
  For each $j' \in S_{t-1}^C$, set the will-reduced value $\Delta_{j'}$.
  Set entering candidate $j^{\text{in}} := \arg\min_{j'\in S_{t-1}}\Delta_{j'}$
  if $\Delta_{j^{\text{in}}} > \Delta_{j^{\text{out}}}$ **then**
    Set $S_t := S_{t-1} - \{j^{\text{out}}\} + \{j^{\text{in}}\}$.
    By eigenvalue decomposition, $[\boldsymbol{A}^{1/2}]_{S_t}[\boldsymbol{A}^{1/2}]_{S_t}^\top = \boldsymbol{U}_{S_t}\boldsymbol{\Lambda}_{S_t}\boldsymbol{U}_{S_t}^\top$.
    Set $[\boldsymbol{V}]_{S_t} = [\boldsymbol{U}_{S_t}]_{[k],[r]}$.
  else
    Break while loop.
  end if
end while

In model-free case,

Theorem 2. Algorithm 1 is a monotone decreasing algorithm in the objective value of SPCA-lasso, i.e., monotone increasing with respect to SPCA.

Theorem 3. Algorithm 1 terminates in at most $\binom{d}{k}$ epochs.

Although this is not the main contribution of our paper, we demonstrate that when additional statistical assumptions/conditions (listed in Appendix) hold, Algorithm 1 combined with a specific initialization method guarantees the following property:

Theorem 4. The primal feasible solution $\boldsymbol{V}_{\text{algo}}$ obtained from Algorithm 1 satisfies $\text{Tr}(\boldsymbol{V}_{\text{algo}}^\top\boldsymbol{\Sigma}\boldsymbol{V}_{\text{algo}}) \ge \text{opt}^{\mathcal{F}}(\boldsymbol{\Sigma}) - 2r\epsilon$ with high probability for any $\epsilon > 0$.

2.4. Reducing the Running Time of SOCIP

In practice, we want to reduce the running time of SOCIP. Here are the techniques that we used to enhance the efficiency in practice.

Threshold: The first technique is to reduce the number of SOS-II constraints. Let ϕ be a threshold parameter that

splits the eigenvalues $\{\lambda_j\}_{j=1}^d$ of sample covariance matrix $\boldsymbol{A}$ into two parts $J^+ = \{j : \lambda_j > \phi\}$ and $J^- = \{j : \lambda_j \le \phi\}$. The objective function $\mathrm{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right)$ then can be represented as

$$\sum_{j\in J^+}(\lambda_j-\phi)\sum_{i=1}^r g_{ji}^2 + \sum_{j\in J^-}(\lambda_j-\phi)\sum_{i=1}^r g_{ji}^2 + r\phi,$$

in which the first summation is convex, and the second summation is concave. Since maximizing a concave function is eqivalent to convex optimization, we replace the second part by a new auxiliary variable s, and only construct a piecewise-linear upper approximation for the quadratic terms g_{ji}^2 in the first summation with $j \in J^+$ as,

$$\sum_{j\in J^+}(\lambda_j-\phi)\sum_{i=1}^r g_{ji}^2 - s + r\phi = \mathrm{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right)$$
$$\text{with}\quad \sum_{j\in J^-}\underbrace{(\phi-\lambda_j)}_{\ge 0}\sum_{i=1}^r g_{ji}^2 \le s \qquad \text{(s-var)}$$

a convex constraint. Notice that in practice, the threshold ϕ should not be too large. Otherwise, let $\mathrm{opt}^{\mathcal{F}}$ be the optimal value of SPCA. If $\phi > \mathrm{opt}^{\mathcal{F}}/r$, then taking $\boldsymbol{V} = \boldsymbol{0}_{d\times r}$ as a trivial feasible solution in $\mathcal{C}$ provides a larger objective function. Let $\hat{\boldsymbol{V}} = (\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_r)$ be a primal feasible solution obtained from Algorithm 1. We have $\mathrm{Tr}(\hat{\boldsymbol{V}}^\top \boldsymbol{A}\hat{\boldsymbol{V}})$ be a reasonable lower (primal) bound estimate of $\mathrm{opt}^{\mathcal{F}}$. Thus when the threshold ϕ satisfies $\phi \le \mathrm{Tr}(\hat{\boldsymbol{V}}^\top \boldsymbol{A}\hat{\boldsymbol{V}})/r$, the SOCIP would provide a appropriate upper (dual) bound. Therefore, by setting the threshold parameter ϕ, we can reduce the number of SOS-II constraints from $\mathcal{O}(d \times r)$ to $\mathcal{O}(|J^+| \times r)$.

Cutting Planes: Similar to classical integer programming, we can incorporate additional cutting planes to improve the efficiency. We propose three families of cutting-planes. First family of cutting-planes is obtained as follows: By Bessel inequality, since $\|\boldsymbol{V}\|_0 \le k$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ are orthogonal, then we obtain that:

$$\sum_{i=1}^r g_{ji}^2 = \sum_{i=1}^r (\boldsymbol{a}_j^\top \boldsymbol{v}_i)^2 = \boldsymbol{a}_j^\top \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{a}_j \le \theta_j^2. \qquad \text{(sparse)}$$

Second family of cutting-planes is obtained as follows: suppose we solve SOCIP to optimality, and let $\boldsymbol{V}^* = (\boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_r^*)$ be its optimal solution, then for each $\boldsymbol{v}_i^*$ with $i \in [r]$, we have $(\boldsymbol{v}_i^*)^\top \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{v}_i^* \le [\boldsymbol{v}_i^*]_{j_1}^2 + \cdots + [\boldsymbol{v}_i^*]_{j_k}^2$.

The third type of cutting plane is based on the property: for any positive seim-definite matrix, the sum of its diagonal entries are equal to the sum of its eigenvalues. Let $[\boldsymbol{A}]_{j_1,j_1}, \ldots, [\boldsymbol{A}]_{j_k,j_k}$ be the largest k diagonal entries of the sample covariance matrix $\boldsymbol{A}$, we have

Proposition 2.5. The objective function in SOCIP is upper bounded by

$$\sum_{j=1}^d \lambda_j \sum_{i=1}^r \xi_{ji} \le [\boldsymbol{A}]_{j_1,j_1} + \cdots + [\boldsymbol{A}]_{j_k,j_k} + \sum_{j=1}^d \frac{r\lambda_j\theta_j^2}{4N^2}$$

when the splitting points $\{\gamma_{ji}^\ell\}_{\ell=-N}^N$ in SOS-II are set to be $\gamma_{ji}^\ell = \frac{\ell}{N}\cdot\theta_j$. Moreover, we have:

$$\begin{aligned}&\textstyle\sum_{j\in J^+}(\lambda_j-\phi)\sum_{i=1}^r \xi_{ji} - s + r\phi \\ &\le [\boldsymbol{A}]_{j_1,j_1} + \cdots + [\boldsymbol{A}]_{j_k,j_k} + \textstyle\sum_{j\in J^+}\frac{r(\lambda_j-\phi)\theta_j^2}{4N^2}\end{aligned} \qquad \text{(cut)}$$

as a cutting plane for the objective function with the threshold function.

Symmetry-breaking Constraints: Note that for any feasible solution $\boldsymbol{V} \in \mathbb{R}^{d\times r}$ of SPCA, permuting the columns of $\boldsymbol{V}$ still guarantees a feasible optimal solution with same objective value. We can use this symmetric property to tighten the feasible region and improve the efficiency. We sort the columns of $\boldsymbol{V}$ by their corresponding k-sparse eigenvalues in decreasing order such that $\boldsymbol{v}_i^\top \boldsymbol{A}\boldsymbol{v}_i \ge \boldsymbol{v}_{i+1}^\top \boldsymbol{A}\boldsymbol{v}_{i+1}$ holds for $i = 1, \ldots, r-1$. However, such constraints are still non-convex, to transfer into convex constraints, we relax the left-hand-side by variables ξ_{ji} for $i = 1, \ldots, r-1$,

$$\textstyle\sum_{j=1}^d \lambda_j \xi_{ji} \ge \boldsymbol{v}_{i+1}^\top \boldsymbol{A}\boldsymbol{v}_{i+1}. \qquad \text{(sym-1)}$$

Similarly, for any column $\boldsymbol{v}$ of a feasible solution $\boldsymbol{V}$, note that flipping the sign of $\boldsymbol{v}$ does not influence its feasibility or its objective value and hence we have for $i = 1, \ldots, r$,

$$\textstyle\sum_{j=1}^d [\boldsymbol{v}_i]_j \ge 0, \qquad \text{(sym-2)}$$

to tighten the feasible region.

Implemented Version of SOCIP: Given ϕ a threshold parameter, and the size of $|J^+|$, set the revised piecewise linear upper approximation (PLA') constraints as,

$$\underbrace{\left\{(g,\xi,\eta):\ \begin{array}{l} g_{ji} = \boldsymbol{a}_j^\top \boldsymbol{v}_i,\ (j,i)\in[d]\times[r] \\ g_{ji} = \sum_{\ell=-N}^N \gamma_{ji}^\ell \eta_{ji}^\ell,\ (j,i)\in J^+\times[r] \\ \xi_{ji} = \sum_{\ell=-N}^N \left(\gamma_{ji}^\ell\right)^2 \eta_{ji}^\ell \\ \left(\eta_{ji}^\ell\right)_{\ell=-N}^N \in \text{SOS-II}\end{array}\right\}}_{=:\text{PLA}'}$$

Thus the implemented version of SOCIP is

$$\begin{array}{ll}\max & \sum_{j\in J^+}(\lambda_j-\phi)\sum_{i=1}^r \xi_{ji} - s + r\phi \\ \text{s.t} & \boldsymbol{V}\in\mathcal{C}',\ (g,\xi,\eta)\in\text{PLA}' \\ & \text{(s-var), (sparse), (cut), (sym-1), (sym-2)}\end{array} \qquad \text{(SOCIP-impl)}$$

Theorem 5. Given the number of splitting point N, the threshold parameter ϕ, and the size of $|J^+|$, the SOCIP-impl can be solved to optimality within polynomial time.

3. Main Numerical Results

Baselines: In this section, we compare the upper bounds obtained from SOCIP-impl against two baselines (Baseline-1, Baseline-2) as defined below.

$$\begin{aligned}\text{Baseline-1} := \quad & [\boldsymbol{A}]_{j_1,j_1} + \cdots + [\boldsymbol{A}]_{j_k,j_k}, \\ & \text{where } [\boldsymbol{A}]_{j_1,j_1} \geq [\boldsymbol{A}]_{j_2,j_3} \geq \cdots [\boldsymbol{A}]_{j_d,j_d} \\ \text{Baseline-2} := \quad & \max_{\boldsymbol{P}} \ \mathrm{Tr}\left(\boldsymbol{A}\boldsymbol{P}\right), \\ & \text{s.t.} \quad \boldsymbol{I}_d \succeq \boldsymbol{P} \succeq \boldsymbol{0}, \\ & \qquad \mathrm{Tr}(\boldsymbol{P}) = r, \ \boldsymbol{1}^\top |\boldsymbol{P}| \boldsymbol{1} \leq rk.\end{aligned}$$

Regarding Baseline-1, since the sum of $[\boldsymbol{A}]_{j_1,j_1}, \ldots, [\boldsymbol{A}]_{j_k,j_k}$ is equal to sum of eigenvalues of sub-matrix indexed by $\{j_1, \ldots, j_k\}$ in $\boldsymbol{A}$, then Baseline-1 can be viewed as an upper bound for the optimal value of SPCA. Moreover, Baseline-1 is tight when we have $r = k$.

Baseline-2 is a SDP relaxation of SPCA by lifting the variables $\boldsymbol{V}$ into its product space $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^\top$.

The numerical results are implemented on two types of instances.

Artificial Instance: The first type of instances are generated artificially from the similar idea of *spiked covariance matrix*: Let d be the size of the covariance matrix with $d \geq 2 \times k$. Set $k = 5$ and support set $S = \{1, 2, 3, 4, 5\}$ with $|S| = k$. Let $\boldsymbol{u}_1, \boldsymbol{u}_2$ be two unit orthogonal vectors with support on S, e.g.,

$$\boldsymbol{u}_1^\top = \Big(\underbrace{\frac{1}{\sqrt{5}}, \frac{1}{\sqrt{5}}, \ldots, \frac{1}{\sqrt{5}}}_{\text{top 5 entries}}\Big), \quad \boldsymbol{u}_2^\top = \Big(\underbrace{\frac{1}{2}, \frac{-1}{2}, \frac{1}{2}, \frac{-1}{2}}_{\text{top 4 entries}}\Big).$$

Set the true covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ be a block spiked covariance matrix (define in appendix) as $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 \oplus \boldsymbol{\Sigma}_2 \oplus \boldsymbol{I}_{d-10}$ with $\boldsymbol{\Sigma}_1 = 55\boldsymbol{u}_1\boldsymbol{u}_1^\top + 52\boldsymbol{u}_2\boldsymbol{u}_2^\top \in \mathbb{R}^{5\times 5}$, $\boldsymbol{\Sigma}_2 = 50 \cdot \boldsymbol{I}_5$. Generate M i.i.d. random samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M \sim N(\boldsymbol{0}_d, \boldsymbol{\Sigma})$, and set the sample covariance matrix $\boldsymbol{A} = \frac{1}{M}\left(\boldsymbol{x}_1\boldsymbol{x}_1^\top + \cdots + \boldsymbol{x}_M\boldsymbol{x}_M^\top\right)$. Notice that: when $k = 5$, for covariance matrix $\boldsymbol{\Sigma}$, the optimal support set for $r = 2$, i.e., $S = \{1, 2, 3, 4, 5\}$ is distinct from the support set for $r = 5$, i.e., $S = \{6, 7, 8, 9, 10\}$.

Real Instance: The second type of instances are real instances. The first two biological data sets (Eisen-1, Eisen-2) [$d \leq 300$] are collected from **?**. The Colon cancer data set [$d = 500$] is from **?**. The Lymphoma data set [$d = 500$] is from **?**. The final instance two real instances is collected from Reddit [$d = 1000, 2000$].

Software & Hardware: All numerical experiments are implemented on MacBookPro13 with 2GHz Intel Core i5 CPU and 8GB 1867MHz LPDDR3 Memory. The SOCIP-impl model was solved using Gurobi 7.0.2.

We measure the performances of SOCIP-impl and Baselines based on the primal-dual gap, defined as Gap := $\frac{\text{ub}-\text{lb}}{\text{lb}}$, where ub denotes the upper bounds (dual bound) obtained from SOCIP-impl or Baselines, and lb denotes the lower bound (primal bound) computed from a feasible solution.

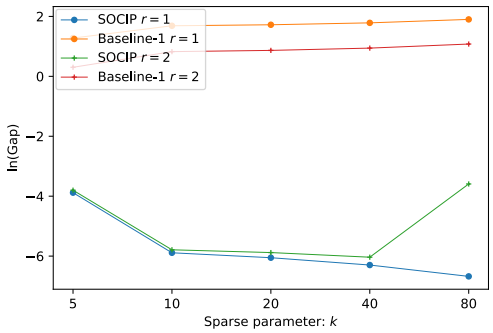

Figure 2. Comparison between SOCIP-impl and Baseline on artificial instance blocked spiked covariance matrix (of size $d = 500$) $r = 2, 3$ for distinct k.

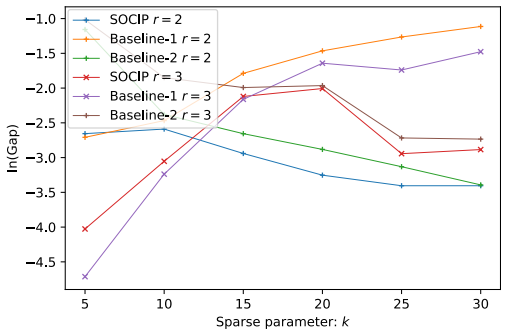

Figure 3. Comparison between SOCIP-impl and Baseline on Eisen-1 (of size $d = 79$) with $r = 2, 3$ for distinct k.

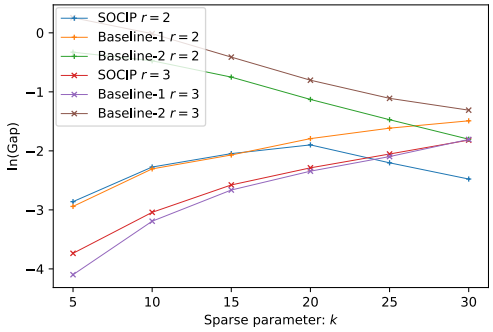

Figure 4. Comparison between SOCIP-impl and Baseline on Eisen-2 (of size $d = 118$) with $r = 2, 3$ for distinct k.

Recall that the Gap we obtained is an upper approximation of Δ mention in Proposition 1.1. All numerical results are reported in Figure 2, 3, 4, 5, 6, 7, 8. Based on the numerical results, we draw the following:

Methods in Artificial Instances: The artificial instances are generated using similar ideas as spiked covariance ma-

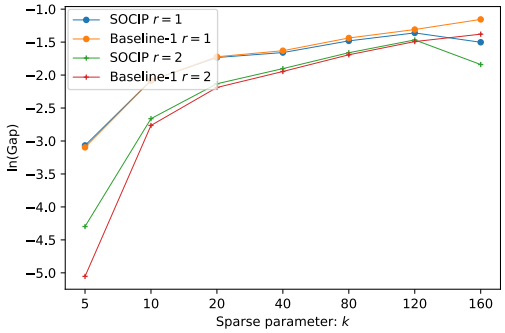

Figure 5. Comparison between SOCIP-impl and Baseline on colon cancer data set (of size $d = 500$) with $r = 1, 2$ for distinct k.

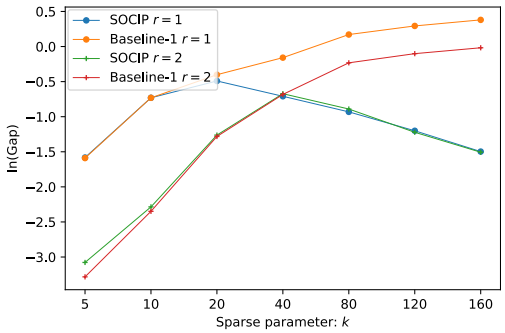

Figure 6. Comparison between SOCIP-impl and Baseline on Lymphoma data set (of size $d = 500$) with $r = 1, 2$ for distinct sparse parameter k.

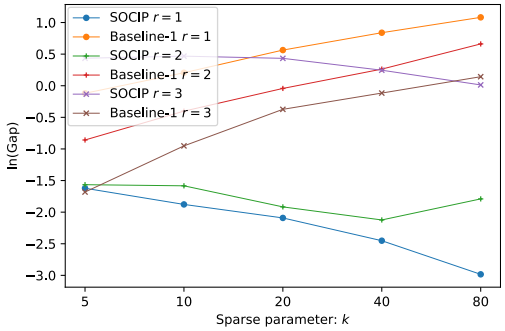

Figure 7. Comparison between SOCIP-impl and Baseline on Reddit data set (of size $d = 1000$) with $r = 1, 2, 3$ for distinct k.

trix. In this instance, we can observe that our SOCIP-impl performs much better than the Baseline-1. For Baseline-2, because the lifting step of SDP relaxation generates too many variables, based on the limitation of hardware, this method fails to obtain any solution or bounds.

Methods in Real Instances: For Eisen-1 & Eisen-2, the performances of SOCIP-impl is significantly better than Baseline-1 and Baseline-2. For colon & Lymphoma with $d = 500$, Baseline-2 is out of memory; SOCIP-impl

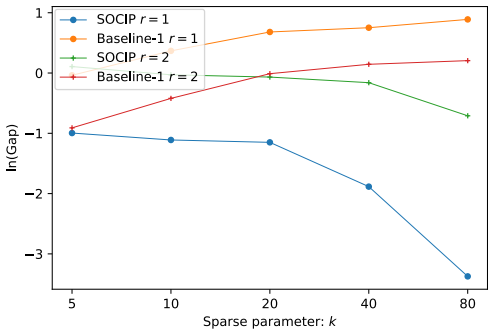

Figure 8. Comparison between SOCIP-impl and Baseline on Reddit data set (of size $d = 2000$) with $r = 1, 2$ for distinct k.

and Baseline-1 perform almost similarly when k is relative small, but as k increases, the results from SOCIP-impl become better than Baseline-1 which demonstrates the scalable property of our method. For Reddit data with $d = 1000, 2000$, similar to the colon & Lymphoma instance, SOCIP-impl performs better as k increases.

4. Conclusion

In this paper, we proposed a monotonically improving heuristic for SPCA problem. We showed that the solution produced by this algorithm are of very high quality by comparing the objective value of the solutions generated to upper bounds. These upper bounds where obtained using second order cone IP relaxation designed in this paper. We also presented theoretical guarantees (affine guarantee) on the quality of the upper bounds produced by the second order cone IP, and its stronger version – semi-definition convex IP. Overall, we have presented a complete solution procedure of generating good solutions and proving quality of these solutions for SPCA. To the best of our knowledge, there is no comparable theoretical or computational results for solving model-free SPCA.

Acknowledgement

We would like to thank the meta-reviewer and three anonymous reviewers for their constructive comments that helped improve the presentation of this paper.