A convex integer programming approach for optimal sparse PCA

Santanu S. Dey* [1], Rahul Mazumder†[2], and Guanyi Wang‡[1]

[1]School of Industrial and Systems Engineering, Georgia Institute of Technology, USA
[2]Operations Research Center, Massachusetts Institute of Technology, USA

October 21, 2018

Abstract

Principal component analysis (PCA) is one of the most widely used dimensionality reduction tools in scientific data analysis. The PCA direction, given by the leading eigenvector of a covariance matrix, is a linear combination of all features with nonzero loadings—this impedes interpretability. Sparse principal component analysis (SPCA) is a framework that enhances interpretability by incorporating an additional sparsity requirement in the feature weights (factor loadings) while finding a direction that explains the maximal variation in the data. However, unlike PCA, the optimization problem associated with the SPCA problem is NP-hard. While many heuristic algorithms based on variants of the power method are used to obtain good solutions, they do not provide certificates of optimality on the solution-quality via associated dual bounds. Dual bounds are available via standard semidefinite programming (SDP) based relaxations, which may not be tight and the SDPs are difficult to scale using off-the-shelf solvers. In this paper, we present a convex integer programming (IP) framework to solve the SPCA problem to near-optimality, with an emphasis on deriving associated dual bounds. We present worst-case results on the quality of the dual bound provided by the convex IP. We empirically observe that the dual bounds are significantly better than worst-case performance, and are superior to the SDP bounds on some real-life instances. Moreover, solving the convex IP model using commercial IP solvers appears to scale much better that solving the SDP-relaxation using commercial solvers. To the best of our knowledge, we obtain the best dual bounds for real and artificial instances for SPCA problems involving covariance matrices of size up to 2000×2000.

1 Introduction

Principal component analysis (PCA) is one of the most widely used dimensionality reduction methods in data science. Given a data matrix $Y \in \mathbb{R}^{m\times n}$ (with m samples and n features; and each feature is centered to have zero mean), PCA seeks to find a principal component (PC) direction $x \in \mathbb{R}^n$ with $\|x\|_2 = 1$ that maximizes the variance of a weighted combination of features. Formally, this PC direction can be found by solving

$$\max_{\|x\|_2=1} x^\top Ax \tag{PCA}$$

*santanu.dey@isye.gatech.edu
†rahulmaz@mit.edu
‡gwang93@gatech.edu

where $A \triangleq \frac{1}{m} Y^\top Y$ is the sample covariance matrix. An obvious drawback of PCA is that all the entries of $\hat{x}$ (an optimal solution to (PCA)) are (usually) nonzero, which leads to the PC direction being a linear combination of all features – this impedes interpretability [8, 19, 36]. In biomedical applications for example, when Y corresponds to the gene-expression measurements for different samples, it is desirable to obtain a PC direction which involves only a handful of the features (e.g, genes) for interpretation purposes. In financial applications (e.g, A may denote a covariance matrix of stock-returns), a sparse subset of stocks that are responsible for driving the first PC direction may be desirable for interpretation purposes. Indeed, in many scientific and industrial applications [31, 1, 16], for additional interpretability, it is desirable for the factor loadings to be sparse, i.e., few of the entries in $\hat{x}$ are nonzero and the rest are zero. This motivates the notion of a sparse principal component analysis (SPCA) [19, 16], wherein, in addition to maximizing the variance, one also desires the direction of the first PC to be sparse in the factor loadings. The most natural optimization formulation of this problem, modifies criterion PCA with an additional sparsity constraint on x leading to:

$$\lambda^k(A) \triangleq \max_{\|x\|_2=1, \|x\|_0 \leq k} x^\top A x \tag{SPCA}$$

where $\|x\|_0 \leq k$, is equivalent to allowing at most k components of x to be nonzero. Unlike the PCA problem, the SPCA problem is NP-hard [23, 9].

Many heuristic algorithms have been proposed in the literature that use greedy methods [19, 35, 17, 14], alternating methods [33] and the related power methods [20]. However, conditions under which (some of) these computationally friendlier methods can be shown to work well, make very strong and often unverifiable assumptions on the problem data. Therefore, the performance of these heuristics (in terms of how close they are to an optimal solution of the SPCA problem) on a given dataset is not clear.

Since SPCA is NP-hard, there has been exciting work in the statistics community [4, 30] in understanding the statistical properties of convex relaxations (e.g., those proposed by [10] and variants) of SPCA. It has been established [4, 30] that the statistical performance of estimators available from convex relaxations are sub-optimal (under suitable modeling assumptions) when compared to estimators obtained by (optimally) solving SPCA—this further underlines the importance of creating tools to be able to solve SPCA to optimality.

Our main goal in this paper is to propose an integer programming framework that allows the computation of good solutions to the SPCA problem with associated certificates of optimality via dual bounds, which make limited restrictive/unverifiable assumptions on the data. To the best of our knowledge, the only published methods for obtaining duals bounds of SPCA are based on semidefinite programming (SDP) relaxations [12, 14, 34, 13] (see Appendix A for the SDP relaxation) and spectral methods involving a low-rank approximation of the matrix A [25]. Both these approaches however, have some limitations. The SDP relaxation does not appear to scale easily (using off-the-shelf solvers) for matrices with more than a few hundred rows/columns, while applications can be significantly larger. Indeed, even a relatively recent implementation based on the Alternating Direction Method of Multipliers for solving the SDP considers instances with $n \approx 200$ [22]. The spectral methods involving a low-rank approximation of A proposed in [25] have a running time of $\mathcal{O}(n^d)$ where d is the rank of the matrix—in order to scale to large instances, no more than a rank 2 approximation of the original matrix seems possible. The paper [3] presents a specialized branch and bound solver[1] to obtain solutions to the SPCA problem, but their method

[1]This paper is not available in the public domain at the time of writing this paper.

can handle problems with $n \approx 100$ – the approach presented here is different, and our proposal scales to problem instances that are much larger.

The methods proposed here are able to obtain approximate dual bounds of SPCA by solving convex integer programs and a related perturbed version of convex integer programs that are easier to solve. The dual bounds we obtain are incomparable to dual bounds based on the SDP relaxation, i.e. neither dominates the other, and the method appears to scale well to matrices up to sizes of 2000×2000.

2 Main results

Given a set S, we denote $\text{conv}(S)$ as the convex hull of S; given a positive integer n we denote $\{1, \ldots, n\}$ by $[n]$; given a matrix A, we denote its trace by $\text{tr}(A)$.

Notice that the constraint $\|x\|_2 = 1, \|x\|_0 \leq k$ implies that $\|x\|_1 \leq \sqrt{k}$. Thus, one obtains the so-called ℓ_1-norm relaxation of SPCA:

$$\text{OPT}_{\ell_1} \triangleq \max_{\|x\|_2 \leq 1, \|x\|_1 \leq \sqrt{k}} x^\top A x. \qquad (\ell_1\text{-relax})$$

The relaxation ℓ_1-relax has two advantages: (a) As shown in Theorem 1 below, ℓ_1-relax gives a constant factor bound on SPCA, (b) The feasible region is convex and all the nonconvexity is in the objective function. We build on these two advantages: our convex IP relaxation is a further relaxation of ℓ_1-relax (together with some implied linear inequalities for SPCA) which heavily use the fact that the feasible region of ℓ_1-relax is convex.

We note that ℓ_1-relax is an important estimator in its own right [31, 16]—it is commonly used in the statistics/machine-learning community as one that leads to an eigenvector of A with entries having a small ℓ_1-norm (as opposed to a small ℓ_0-norm). However, we emphasize that ℓ_1-relax is a nonconvex optimization problem—globally optimizing ℓ_1-relax is challenging—we show in this paper, how one can use IP methods to solve ℓ_1-relax to certifiable optimality.

The rest of this section is organized as follows: In Section 2.1, we present the constant factor bound on SPCA given by ℓ_1-relax, improving upon some known results. In Section 2.2, we present the construction of our convex IP and prove results on the quality of bound provided. In Section 2.3, we discuss perturbing the original matrix in order to make the convex IP more efficiently solvable while still providing reasonable dual bounds. In Section 4, we present results from our computational experiments.

2.1 Quality of ℓ_1-relaxation as a surrogate for the SPCA problem

The following theorem is an improved version of a result appearing in [29] (Exercise 10.3.7).

Theorem 1. *The objective value OPT_{ℓ_1} is upper bounded by a multiplicative factor ρ^2 away from $\lambda^k(A)$, i.e., $\lambda^k(A) \leq OPT_{\ell_1} \leq \rho^2 \cdot \lambda^k(A)$ with $\rho \leq 1 + \sqrt{\frac{k}{k+1}}$.*

Proof of Theorem 1 is provided in Section 3. While we have improved upon the bound presented in [29], we do not know if this new bound is tight.

Theorem 1 has implications regarding existence of polynomial-time algorithms to obtain a constant-factor approximation guarantee for ℓ_1-relax. In particular, the proof of Theorem 1 implies that if one can obtain a solution for ℓ_1-relax which is within a constant factor, say θ, of OPT_{ℓ_1},

then a solution for SPCA problem can be obtained, which is within a constant factor (at most $\theta\rho \approx 2\theta$) of $\lambda^k(A)$. Therefore, the ℓ_1-relax problem is also inapproximable in general.

2.2 Convex integer programming method

A classical integer programming approach to solving SPCA would be to go to an extended space involving the product of x-variables and include one binary variable per x-variable in order to model the ℓ_0-norm constraint, resulting in a very large number of binary variables. In particular, a typical model could be of the form:

$$\begin{aligned}
\max \quad & \operatorname{tr}(AX) && (1)\\
\text{s.t.} \quad & x_i \le z_i,\ i \in [n] && (2)\\
& \sum_{j=1}^{n} z_i \le k && (3)\\
& \|x\|_2 \le 1 && (4)\\
& \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0 && (5)\\
& \operatorname{rank}\left(\begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \right) = 1 && (6)\\
& z \in \{0,1\}^n. && (7)
\end{aligned}$$

It is easy to see that such a model is challenging due to (a) "quadratic" increase in number of variables (X) (b) the presence of the rank constraint and (c) n binary variables. Even with significant progress, it is well-known that solving such problems beyond n being a few hundred variables is extremely challenging [5, 15]. Indeed, solving instances with arbitrary quadratic objective and bound constraints cannot be solved (exactly) by modern state-of-the-art methods as soon as the number of variables exceed a hundred or so [7, 6].

On the other hand, as mentioned before, the feasible region of ℓ_1-relax is a convex set. Therefore, we do not have to include binary variables as in the case for modeling the ℓ_0-norm constraint. Moreover, Theorem 1 suggests that ℓ_1-relax will provide quite strong dual bounds for SPCA. Thus, we will use ℓ_1-relax as our basic relaxation — we note that the factor of $1+\sqrt{\frac{k}{k+1}}$ is a worst case bound, and as we see in our numerical experiments the objective function values of the two problems are quite close.

Since the feasible region of ℓ_1-relax is a convex set we need to model/upper bound the objective function using convex IP techniques. We follow the following arguments:

1. By a spectral decomposition, let $A = \sum_{i=1}^{n} \lambda_i v_i v_i^\top$ where $(\lambda_i)_{i=1}^n, (v_i)_{i=1}^n$ are unit norm orthogonal eigen-pairs. Then the objective function of ℓ_1-relax is:

$$\sum_{i=1}^{n} \lambda_i (x^\top v_i)^2.$$

2. Assuming that $\lambda \le \lambda^k(A)$, we have that $x^\top A x = x^\top (A - I\lambda)x + \lambda$ for x such that $\|x\|_2 = 1$, where I is the identity matrix. Therefore, if we split the eigenvalues into two sets as $\{i : \lambda_i >$

$\lambda\}$ and $\{i : \lambda_i \le \lambda\}$, the objective function can be represented as

$$\lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)(x^\top v_i)^2 + \sum_{i\in\{i:\lambda_i\le\lambda\}} (\lambda_i - \lambda)(x^\top v_i)^2.$$

3. For each index $i \in \{i : \lambda_i > \lambda\}$, we replace $x^\top v_i$ with a single continuous variable g_i, and set $\theta_i \leftarrow \max\{x^\top v_i : \|x\|_2 \le 1, \|x\|_0 \le k\}$ (or $\theta_i \leftarrow \max\{x^\top v_i : \|x\|_2 \le 1, \|x\|_1 \le \sqrt{k}\}$ if we explicitly want a relaxation of ℓ_1-relax) as an upper bound of g_i. Then for each g_i with $i \in \{i : \lambda_i > \lambda\}$, we construct a piecewise linear upper approximation ξ_i for g_i^2. Such a piecewise linear upper approximation is usually modeled via SOS-2 constraints [24], and this seems to work well in our numerical experiments.

4. For the term $\sum_{i\in\{i:\lambda_i\le\lambda\}}(\lambda_i - \lambda)(x^\top v_i)^2$, since $\lambda_i - \lambda \le 0$, we obtain a convex constraint $\sum_{i\in\{i:\lambda_i\le\lambda\}} -(\lambda_i - \lambda)(x^\top v_i)^2 \le s$.

Therefore, a convex integer programming problem is obtained as follows:

$$\begin{aligned}
&\lambda + \max_{x,y,g,\xi,\eta,s} \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\xi_i - s \qquad (\triangleq \mathrm{OPT}_{\text{convex-IP}}) \\
&\text{s.t.} \quad
\begin{cases}
g_i = x^\top v_i,\ i \in [n] & (x^\top A x = \sum_{i=1}^n \lambda_i g_i^2) \\
\begin{cases} g_i = \sum_{j=-N}^{N} \gamma_i^j \eta_i^j,\ i \in \{i : \lambda_i > \lambda\} \\ \xi_i = \sum_{j=-N}^{N} (\gamma_i^j)^2 \eta_i^j,\ i \in \{i : \lambda_i > \lambda\} \\ (\eta_i^{-N}, \dots, \eta_i^N) \in \text{SOS-2},\ i \in \{i : \lambda_i > \lambda\} \end{cases} & ((x^\top v_i)^2 \le \xi_i,\ i \in \{i : \lambda_i > \lambda\}) \\
\begin{cases} \sum_{i=1}^n x_i^2 \le 1 \\ \sum_{i\in\{i:\lambda_i>\lambda\}} \xi_i + \sum_{i\in\{i:\lambda_i\le\lambda\}} g_i^2 \le 1 + \frac{1}{4N^2} \sum_{i\in\{i:\lambda_i>\lambda\}} \theta_i^2 \end{cases} & (\|x\|_2 \le 1) \\
\sum_{i=1}^n y_i \le \sqrt{k} \text{ and } y_i \ge x_i, y_i \ge -x_i\ i \in \{i : \lambda_i > \lambda\} & (\|x\|_1 \le \sqrt{k}) \\
-\theta_i \le g_i \le \theta_i,\ i \in [n] & (|g_i| \le \theta_i,\ i \in [n]) \\
\sum_{i\in\{i:\lambda_i\le\lambda\}} -(\lambda_i - \lambda) g_i^2 \le s &
\end{cases}
\end{aligned} \tag{Convex-IP}$$

where, for each $i \in \{i : \lambda_i > \lambda\}$, let $2N + 1$ be the number of splitting points that partition the domain of g_i, i.e., $[-\theta_i, \theta_i]$ equally, and let $(\gamma_i^j)_{j=-N}^N \leftarrow \left(\frac{j}{N} \cdot \theta_i\right)_{j=-N}^N$ be the value of the j^{th} splitting point.

Since v_i's are orthogonal, $\sum_{i=1}^n x_i^2 \le 1$ implies $\sum_{i=1}^n g_i^2 \le 1$. Then together with ξ_i representing g_i^2, we can obtain the implied inequality $\sum_{i\in\{i:\lambda_i>\lambda\}} \xi_i + \sum_{i\in\{i:\lambda_i\le\lambda\}} g_i^2 \le 1 + \frac{1}{4N^2} \sum_{i\in\{i:\lambda_i>\lambda\}} \theta_i^2$. The extra term in the right-hand-side reflects the fact that ξ_i is not exactly equal to g_i^2, but only a piecewise linear upper bound on g_i^2. In fact, we have that $\xi_i \le g_i^2 + \frac{\theta_i}{4N^2}$. This constraint (cutting-plane) helps in improving the dual bound at the root node and significantly improves the running time of the solver. We arrive at the following result:

Proposition 1. *The optimal objective value $OPT_{convex\text{-}IP}$ of Convex-IP is an upper bound on the SPCA problem.*

Proposition 1 is formally verified in Appendix B.

Next combining the result of Theorem 1 with the quality of the approximation of the objective function of ℓ_1-relax by Convex-IP, we obtain the following result:

Proposition 2. *The optimal objective value $OPT_{convex\text{-}IP}$ of Convex-IP is upper bounded by*

$$OPT_{convex\text{-}IP} \leq \rho^2 \lambda^k(A) + \frac{1}{4N^2} \sum_{i \in \{i:\lambda_i > \lambda\}} (\lambda_i - \lambda)\theta_i^2.$$

A proof of Proposition 2 is presented in Appendix C.

Finally, let us discuss why we expect Convex-IP to be appealing from a computational viewpoint. Unlike typical integer programming approaches, the number of binary variables in Convex-IP is $(2N+1) \cdot |\{i : \lambda_i > \lambda\}|$ which is usually significantly smaller than n. Indeed, heuristics for SPCA generally produce good values of λ, and in almost all experiments we found that $|\{i : \lambda_i > \lambda\}| \ll n$. Moreover, N is a parameter we control. In order to highlight the "computational tractability" of Convex-IP, we formally state the following result:

Proposition 3. *Given the number of splitting points N and the size of set $\{i : \lambda_i > \lambda\}$, the Convex-IP problem can be solved in polynomial time.*

Note that the convex integer programming method which is solvable in polynomial time, does not contradict the inapproxamability of the SPCA problem, since $\text{OPT}_{\text{convex-IP}}$ is upper bounded by the sum of $\rho^2\lambda^k(A)$ and a term corresponding to the sample covariance matrix.

2.3 Improving the running time of Convex-IP

2.3.1 Perturbation of the covariance matrix A:

Empirically, we use a (sequence of) perturbations on the covariance matrix A to reduce the running time of solving the convex IP. Recall that λ denotes a lower bound of $\lambda^k(A)$.

1. Set $\bar{\lambda} \triangleq \max\{\lambda_i : \lambda_i \leq \lambda\}$ (where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of A). We assume $\bar{\lambda} < \lambda$. However, when $\bar{\lambda} \triangleq \max\{\lambda_i : \lambda_i \leq \lambda\} = \lambda$, one can apply Algorithm 1 to obtain a matrix $\bar{A} \succeq A$, such that $\bar{\lambda} < \lambda$. We then replace A by $\bar{A}$.

Algorithm 1 Perturbation of A

Input: Sample covariance matrix A and λ.
Output: A perturbed sample covariance matrix $\bar{A}$ with distinct eigenvalues such that $\bar{A} \succeq A$.
function PERTURBATION METHOD(A, λ)
    Compute spectral decomposition on A as $A = V^\top \Lambda V$ and let $\lambda_1 > \cdots > \lambda = \lambda_j > \cdot\lambda_p \geq 0$ be all its distinct values of eigenvalues where $p \leq n$.
    Set $\Delta\lambda \leftarrow \min\{\lambda_i - \lambda_{i+1} : i = 1, \ldots, p-1\}$.
    Set $\bar{\Lambda} \leftarrow \Lambda + \text{diag}\left(\frac{i-1}{n}\epsilon : i = n, \ldots, 1\right)$ with $\epsilon < \Delta\lambda$.
    return $\bar{A} \leftarrow V^\top \bar{\Lambda} V$.
end function

2. Perturb the covariance matrix $A = \sum_{i=1}^{n} \lambda_i v_i v_i^\top$ by $\bar{A} = \sum_{i\in\{i:\lambda_i>\lambda\}} \lambda_i v_i v_i^\top + \sum_{i\in\{i:\lambda_i\le\lambda\}} \bar{\lambda} v_i v_i^\top$. Note that the objective value $\mathrm{OPT}_{\text{convex-IP}}(\bar{A})$ in Convex-IP is an upper bound on $\mathrm{OPT}_{\text{convex-IP}}(A)$. Replace A by $\bar{A}$.

3. Therefore, the convex constraint $\sum_{i\in\{i:\lambda_i\le\lambda\}} -(\lambda_i - \lambda) g_i^2 \le s$ in Convex-IP can be replaced by $\sum_{i\in\{i:\lambda_i\le\lambda\}} -(\bar{\lambda} - \lambda) g_i^2 \le s$, i.e., $\sum_{i\in\{i:\lambda_i\le\lambda\}} g_i^2 \le \frac{s}{\lambda - \bar{\lambda}}$.

4. The constraint $\sum_{i\in\{i:\lambda_i\le\lambda\}} -(\bar{\lambda} - \lambda) g_i^2 \le s$ is satisfied at equality for any optimal solution, therefore the following convex constraints

$$1 \le \sum_{i\in\{i:\lambda_i>\lambda\}} \bar{\xi}_i + \sum_{i\in\{i:\lambda_i\le\lambda\}} \bar{g}_i^2 \le 1 + \frac{1}{4N^2} \sum_{i\in\{i:\lambda_i>\lambda\}} \theta_i^2,$$
$$\sum_{i=1}^{n} \bar{g}_i^2 = \sum_{i\in\{i:\lambda_i>\lambda\}} \bar{g}_i^2 + \sum_{i\in\{i:\lambda_i\le\lambda\}} \bar{g}_i^2 \le 1,$$

imply the following inequalities:

$$1 - \frac{\bar{s}}{\lambda - \bar{\lambda}} \le \sum_{i\in\{i:\lambda_i>\lambda\}} \bar{\xi}_i \le 1 + \frac{1}{4N^2} \sum_{i\in\{i:\lambda_i>\lambda\}} \theta_i^2 - \frac{\bar{s}}{\lambda - \bar{\lambda}},$$
$$\sum_{i\in\{i:\lambda_i>\lambda\}} \bar{g}_i^2 \le 1 - \frac{\bar{s}}{\lambda - \bar{\lambda}}.$$

Thus a simplified convex IP corresponding to the perturbed version of the matrix is:

$$\lambda + \max_{x,y,g,\xi,\eta,s} \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\xi_i - s \triangleq \mathrm{OPT}_{\text{Pert-Convex-IP}}$$
$$\text{s.t.} \quad \begin{cases} g_i = x^\top v_i,\ i \in \{i : \lambda_i > \lambda\} & (x^\top \bar{A} x = \sum_{i\in I_1} \lambda_i g_i^2 + \sum_{i\in I_1^C} \bar{\lambda} g_i^2) \\ \begin{cases} g_i = \sum_{j=-N}^{N} \gamma_i^j \eta_i^j,\ i \in \{i : \lambda_i > \lambda\} \\ \xi_i = \sum_{j=-N}^{N} (\gamma_i^j)^2 \eta_i^j,\ i \in \{i : \lambda_i > \lambda\} \\ (\eta_i^{-N}, \ldots, \eta_i^N) \in \text{SOS-2},\ i \in \{i : \lambda_i > \lambda\} \end{cases} & ((x^\top v_i)^2 \le \xi_i,\ i \in \{i : \lambda_i > \lambda\}) \\ \sum_{i=1}^{n} x_i^2 \le 1 & \\ \sum_{i\in\{i:\lambda_i>\lambda\}} g_i^2 \le 1 - \frac{s}{\lambda - \bar{\lambda}} & (\|x\|_2 \le 1) \\ 1 - \frac{s}{\lambda - \bar{\lambda}} \le \sum_{i\in\{i:\lambda_i>\lambda\}} \xi_i \le 1 + \frac{1}{4N^2} \sum_{i\in\{i:\lambda_i>\lambda\}} \theta_i^2 - \frac{s}{\lambda - \bar{\lambda}} & \\ \sum_{i=1}^{n} y_i \le \sqrt{k} \text{ and } y_i \ge |x_i|,\ i \in \{i : \lambda_i > \lambda\} & (\|x\|_1 \le \sqrt{k}) \\ -\theta_i \le g_i \le \theta_i,\ i \in \{i : \lambda_i > \lambda\}. & (|g_i| \le \theta_i,\ i \in [n]) \end{cases}$$

(Pert-Convex-IP)

Proposition 4. *The optimal objective value $OPT_{Pert\text{-}Convex\text{-}IP}$ of Pert-Convex-IP is upper bounded by*

$$OPT_{Pert\text{-}Convex\text{-}IP} \leq \rho^2\lambda^k(A) + \rho^2(\bar{\lambda} - \lambda_{\min}(A)) + \frac{1}{4N^2}\sum_{i\in\{i:\lambda_i>\lambda\}}(\lambda_i - \lambda)\theta_i^2.$$

Note that in Pert-Convex-IP, we do not need the variables $g_i, i \in \{i : \lambda_i \leq \lambda\}$ which greatly reduces the number of variables since in general $|\{i : \lambda_i \geq \lambda\}| \ll n$. In practice, we note a significant reduction in running time, while the dual bound obtained from Pert-Convex-IP model remains reasonable. More details are presented in Section 4.

2.3.2 Refining the splitting points

Since the Pert-Convex-IP model runs much faster than the Convex-IP model, we run the Pert-Convex-IP model iteratively. In each new iteration, we add one extra splitting point describing each ξ_i function. In particular, once we solve a Pert-Convex-IP model, we add one splitting point at the optimal value of g_i.

2.3.3 Cutting planes

Proposition 5. *Let $x \in \mathbb{R}^n$. Let $|x_{i_1}| \geq |x_{i_2}| \geq \cdots \geq |x_{i_{n-1}}| \geq |x_{i_n}|$ where $\{i_1, i_2, \ldots, i_k\} \subseteq \{1, \ldots, n\}$. Then let v be the vector:*

$$v_{i_j} = \begin{cases} |x_{i_j}| & \text{if } j \leq k \\ |x_{i_k}| & \text{if } j > k. \end{cases} \tag{8}$$

Also let $b_{(v)} := \|(v_{i_1}, v_{i_2}, v_{i_3}, \ldots, v_{i_k})\|_2$. The inequality

$$v^\top y \leq b_{(v)}, \tag{9}$$

is a valid inequality for SPCA.

The validity of this inequality is clear: If (x, y) is a feasible point, then the support of y is at most k and $\|y\|_2 \leq 1$. Therefore, $v^\top y \leq \|(v_{i_1}, v_{i_2}, v_{i_3}, \ldots, v_{i_k})\|_2 = b_{(v)}$. Notice that this inequality is not valid for ℓ_1-relax. Also see [21].

We add these inequalities at the end of each iteration for the model where the seeding x for constructing v is chosen to be the optimal solution of the previous iteration.

3 Proof of Theorem 1

Given a vector $v \in \mathbb{R}^n$, we denote the j^{th} coordinate as $[v]_j$ in this section. Define

$$S_k \triangleq \{x \in \mathbb{R}^n : \|x\|_2 \leq 1, \|x\|_0 \leq k\}, \tag{10}$$

$$T_k \triangleq \{x \in \mathbb{R}^n : \|x\|_2 \leq 1, \|x\|_1 \leq \sqrt{k}\}. \tag{11}$$

Note that any $x \in T_k$ can be represented as a nonnegative combination of points in S_k, i.e., $x = x^1 + \cdots + x^m$ and $x^i \in S_k$ for all i. Here we think of each x^i as a projection onto some unique

k components of x and setting the other components to *zero*. Let $y^i = \frac{x^i}{\|x^i\|_2}$, then $y^i \in S_k$. Now we have, $x = \sum_{i=1}^m \|x^i\|_2 \cdot y_i$, and therefore

$$\frac{1}{\sum_{i=1}^m \|x^i\|_2} x = \sum_{i=1}^m \frac{\|x^i\|_2}{\sum_{i=1}^m \|x^i\|_2} \cdot y_i. \tag{12}$$

Thus, if we scale $x \in T_k$ by $\|x^1\|_2 + \ldots + \|x^m\|_2$, then the resulting vector belongs to $\text{conv}(S_k)$. Since we want this scaling factor to be as small as possible, we solve the following optimization problem:

$$\min \|x^1\|_2 + \ldots + \|x^m\|_2 : \; x = x^1 + \ldots + x^m; \; x^i \in S_k, \forall i \in [m]. \tag{Bound}$$

Without loss of generality, we assume that $x \geq 0$ and $[x]_1 \geq [x]_2 \geq \cdots \geq [x]_n \geq 0$. Let $x = v^1 + \ldots + v^m$ where $v^1, \ldots, v^m \in S_k$ is an optimal solution of Bound. The following proposition presents a result on an optimal solution of Bound.

Proposition 6. *Let $I^1, \ldots, I^m$ be a collection of supports such that: I^1 indexes the k largest (in absolute value) components in x, I^2 indexes the second k largest (in absolute value) components in x, and so on (note that $m = \lceil \frac{n}{k} \rceil$). Then $I^1, \ldots, I^m$ is an optimal set of supports for Bound.*

Proof. We prove this result by the method of contradiction. Suppose we have an optimal representation as $x = \bar{v}^1 + \cdots \bar{v}^m$ — and without loss of generality, we assume that $\|\bar{v}^1\|_2 \geq \cdots \geq \|\bar{v}^m\|_2$. Let $\bar{I}^1, \ldots, \bar{I}^m$ be the set of supports of $\bar{v}^1, \ldots, \bar{v}^m$ respectively, where we assume that the indices within each support vector are ordered such that

$$[x_{\bar{I}^j}]_1 \geq [x_{\bar{I}^j}]_2 \geq \cdots \geq [x_{\bar{I}^j}]_g$$

for all $j \in \{1, \ldots, m\}$ (note that $g = k$ if $j < m$).

Let $\bar{I}^p$ be the first support that is different from I^p, i.e., $\bar{I}^1 = I^1, \ldots, \bar{I}^{p-1} = I^{p-1}$ and $\bar{I}^p \neq I^p$. Let $[I^p]_q$ be the first index in I^p that does not belong to $\bar{I}^p$ with $q \leq k$ since $\|\bar{I}^p\|_0 = k$. Therefore, $[I^p]_q$ must be in $\bar{I}^{p'}$ where $p' > p$. Note now that by construction of I and our assumption on $\bar{I}$, we have that $[x_{I^p}]_q \geq [x_{\bar{I}^p}]_q \geq [x_{\bar{I}^p}]_k$. Now we exchange the index $[I^p]_q$ in $\bar{I}^{p'}$ with $[\bar{I}^p]_k$ in $\bar{I}^p$. We have:

$$\sqrt{\|x_{\bar{I}^p}\|_2^2 + ([x_{I^p}]_q)^2 - ([x_{\bar{I}^p}]_k)^2} + \sqrt{\|x_{\bar{I}^{p'}}\|_2^2 + ([x_{\bar{I}^p}]_k)^2 - ([x_{I^p}]_q)^2} \leq \|x_{\bar{I}^p}\|_2 + \|x_{\bar{I}^{p'}}\|_2, \tag{13}$$

which holds because $\|x_{\bar{I}^p}\|_2 \geq \|x_{\bar{I}^{p'}}\|_2$ and $([x_{I^p}]_q)^2 - ([x_{\bar{I}^p}]_k)^2 \geq 0$.

Now repeating the above step, we obtain the result. □

Based on Proposition 6, for any fixed $x \in T_k$, we can find out an optimal solution of Bound in closed form. Now we would like to know, for which vector x, the scaling factor $\|v^1\|_2 + \ldots + \|v^m\|_2$ will be the largest. Let ρ be obtained by solving the following optimization problem:

$$\begin{array}{ll} \rho = \max_x & \|x_{I^1}\|_2 + \cdots + \|x_{I^m}\|_2 \\ \text{s.t.} & x = x_{I^1} + \cdots + x_{I^m} \\ & \|x\|_2^2 = \|x_{I^1}\|_2^2 + \cdots + \|x_{I^m}\|_2^2 \leq 1 \\ & \|x\|_1 = \|x_{I^1}\|_1 + \cdots + \|x_{I^m}\|_1 \leq \sqrt{k} \\ & [x]_1 \geq \cdots \geq [x]_n \geq 0. \end{array} \tag{Approximation ratio}$$

Then we obtain

$$T_k \subseteq \rho \cdot \mathrm{Conv}\left(S_k\right). \tag{14}$$

Although the optimal objective value of Approximation ratio is hard to compute exactly, we can still find an upper bound.

Lemma 1. *The objective value ρ of Approximation ratio is bounded from above by $1+\sqrt{\frac{k}{k+1}}$.*

Proof. First consider the case when $n \leq 2k$. In this case, $m \leq 2$. Consider the optimization problem:

$$\begin{aligned} \theta = \max \quad & u+v \\ \text{s.t.} \quad & u^2+v^2 \leq 1 \end{aligned}$$

If we think of $\|x_{I^1}\|_2$ as u and $\|x_{I^2}\|_2$ as v, then we see that the above problem is a relaxation of Approximation ratio and therefore $\theta = \sqrt{2}$ is an upper bound on ρ. Noting that $\sqrt{2} \leq 1+\sqrt{\frac{k}{k+1}}$ for all $k \geq 1$, we have the result.

Now we assume that $n > 2k$ and consequently $m > 2$. From Approximation ratio, let $\|x_{I^1}\|_1 = t$ and $\|x_{I^1}\|_2 = \gamma$. Based on the standard relationship between ℓ_1 and ℓ_2 norm, we have

$$\gamma \leq t \leq \sqrt{k}\gamma.$$

Since each coordinate of x_{I^2} is smaller in magnitude than the average coordinate of x_{I^1}, we have

$$\|x_{I^2}\|_2 \leq \sqrt{\left(\frac{\|x_{I^2}\|_1}{k}\right)^2 k} = \frac{t}{\sqrt{k}}. \tag{15}$$

Also note that an alternative bound is given by

$$\|x_{I^2}\|_2 \leq \sqrt{1-\gamma^2}.$$

Using an argument similar to the one used to obtain (15), we obtain that

$$\sum_{i=3}^{m} \|x_{I^i}\|_2 \leq \sum_{i=2}^{m-1} \sqrt{\left(\frac{\|x_{I^i}\|_1}{k}\right)^2 k} = \frac{1}{\sqrt{k}} \sum_{i=2}^{m-1} \|x_{I^i}\|_1 \leq \frac{\sqrt{k}-t}{\sqrt{k}}.$$

Therefore we obtain

$$\sum_{i=1}^{m} \|x_{I^i}\|_2 = \|x_{I^1}\|_2 + \|x_{I^2}\|_2 + \sum_{i=3}^{m} \|x_{I^i}\|_2 \leq \gamma + \min\left\{\frac{t}{\sqrt{k}}, \sqrt{1-\gamma^2}\right\} + 1 - \frac{t}{\sqrt{k}}. \text{ (Upper-Bound)}$$

Now we consider two cases:

1. If $\frac{t}{\sqrt{k}} \geq \sqrt{1-\gamma^2}$, then Upper-Bound becomes $\gamma+\sqrt{1-\gamma^2}+1-\frac{t}{\sqrt{k}}$. Since $\gamma \geq \frac{t}{\sqrt{k}} \geq \sqrt{1-\gamma^2}$, γ satisfies $\gamma \geq \frac{1}{\sqrt{2}}$. Moreover we have that $t \geq \gamma, t \geq \sqrt{k(1-\gamma^2)}$. Since $\gamma \leq \sqrt{k(1-\gamma^2)}$ iff

$\gamma \leq \sqrt{\frac{k}{k+1}}$ we obtain two cases:

$$\gamma + \sqrt{1-\gamma^2} + 1 - \frac{t}{\sqrt{k}} \leq \begin{cases} \gamma + \sqrt{1-\gamma^2} + 1 - \sqrt{1-\gamma^2} & \text{if } \gamma \in \left[\frac{1}{\sqrt{2}}, \sqrt{\frac{k}{k+1}}\right] \\ \gamma + \sqrt{1-\gamma^2} + 1 - \frac{\gamma}{\sqrt{k}} & \text{if } \gamma \in \left[\sqrt{\frac{k}{k+1}}, 1\right] \end{cases}$$
$$\leq \begin{cases} 1 + \sqrt{\frac{k}{k+1}} \\ 1 + \sqrt{\frac{k}{k+1}} \end{cases} \tag{16}$$

where (i) the first inequality holds when $\gamma = \sqrt{\frac{k}{k+1}}$, (ii) the second inequality holds since the function $f(\gamma) = \gamma + \sqrt{1-\gamma^2} + 1 - \frac{\gamma}{\sqrt{k}}$ achieves (local and global) maximum at point $\gamma = \sqrt{\frac{k+1-2\sqrt{k}}{2k+1-2\sqrt{k}}}$ which is less than $\sqrt{\frac{k}{k+1}}$ for $k = 1, 2, \ldots$, thus $f(\gamma) \leq \max\left\{f\left(\sqrt{\frac{k}{k+1}}\right), f(1)\right\} = 1 + \sqrt{\frac{k}{k+1}}$ for part $\gamma \in \left[\sqrt{\frac{k}{k+1}}, 1\right]$.

2. If $\frac{t}{\sqrt{k}} \leq \sqrt{1-\gamma^2}$, then Upper-Bound becomes $\gamma + 1$. Note now that $\frac{\gamma}{\sqrt{k}} \leq \frac{t}{\sqrt{k}} \leq \sqrt{1-\gamma^2}$, implies that γ satisfies $\gamma \leq \sqrt{\frac{k}{k+1}}$. Therefore, $1 + \gamma \leq 1 + \sqrt{\frac{k}{k+1}}$.

Therefore, this upper bound holds. □

Therefore, we can show Theorem 1 holds.

Proof. Proof. Proof of Theorem 1. Since $T_k \subseteq \rho \cdot \mathrm{Conv}(S_k)$ with $\rho \leq 1 + \sqrt{\frac{k}{k+1}}$ and the objective function is maximizing a convex function, we obtain that $\lambda^k(A) \leq \mathrm{OPT}_{\ell_1} \leq \rho^2 \cdot \lambda^k(A)$. □

4 Numerical experiments

In this section, we report results on our empirical comparison of the performances of Convex-IP method, Pert-Convex-IP method and the SDP relaxation method.

4.1 Hardware and Software

All numerical experiments are implemented on MacBookPro13 with 2 GHz Intel Core i5 CPU and 8 GB 1867 MHz LPDDR3 Memory. Convex-IPs were solved using Gurobi 7.0.2. SDPs were solved using Mosek 8.0.0.60.

4.2 Obtaining primal solutions

We used a heuristic, which is very similar to the truncated power method [33], but has some advantages over the truncated power method. Given $v \in \mathbb{R}^n$, let $I_k(v)$ be the set of indices corresponding to the top k entries of v (in absolute value).

We start with a random initialization x^0 such that $\|x^0\|_2 = 1$, and set $I^0 \leftarrow I_k(V^\top x^0)$ where V is a square root of A, i.e. $A = V^\top V$. In the i^{th} iteration, we update

$$I^i \leftarrow I_k(V^\top x^i),\ x^{i+1} \leftarrow \arg\max_{\|x\|_2=1} x^\top A_{I^i} x \tag{17}$$

where $A_I \in \mathbb{R}^{n\times n}$ is the matrix with $[A_I]_{i,j} = [A]_{i,j}$ for all $i, j \in I$ and $[A_I]_{i,j} = 0$ otherwise. It is easy to see that $x^1, x^2, \ldots$ satisfy the condition $\|x\|_0 \le k$. Moreover, using the fact A is a PSD matrix, it is easy to verify that $(x^{i+1})^\top A x^{i+1} \ge (x^i)^\top A x^i$ for all i. Therefore, in each iteration, the above heuristic method leads to an improved feasible solution for the SPCA problem.

Our method has two clear advantages over the truncated power method:

- We use standard and efficient numerical linear algebra methods to compute eigenvalues of small $k \times k$ matrices.
- The termination criteria used in our algorithm is also simple: if $I^i = I^{i'}$ for some $i' < i$, then we stop. Clearly, this leads to a finite termination criteria.

In practice, we stop using a stopping criterion based on improvement and number of iterations instead of checking $I^i = I^{i'}$. Details are presented in Algorithm 2.

Algorithm 2 Primal Algorithm

Input: Sample covariance matrix A, cardinality constraint k, initial vector x^0.
Output: A feasible solution x^* of SPCA, and its objective value.
function HEURISTIC METHOD(A, k, x^0)
    Start with an initial (randomized) vector x^0 such that $\|x^0\|_2 = 1$ and $\|x^0\|_0 \le k$.
    Set the initial current objective value Obj $\leftarrow (x^0)^\top A x^0$.
    Set the initial past objective value $\tilde{\text{Obj}} \leftarrow 0$.
    Set the maximum number of iterations be $i^{\max}$.
    while Obj $-$ $\tilde{\text{Obj}} > \epsilon$ and $i \le i^{\max}$ **do**
        Set $\tilde{\text{Obj}} \leftarrow$ Obj.
        Set $I^i \leftarrow I_k(V^\top x^i)$.
        Set $x^{i+1} \leftarrow \arg\max_{\|x\|_2=1} x^\top A_{I^i} x$.
        Set Obj $\leftarrow (x^{i+1})^\top A x^{i+1}$.
    end while
    return x^* as the final x obtained from while-loop, and Obj.
end function

We use the values of $\epsilon = 10^{-6}$ and $i^{\max} = 20$ in our experiments in Algorithm 2.

Our Algorithm may also be interpreted as a version of the "alternating method" used regularly as a heuristic for bilinear programs. The sparse PCA problem can be equivalently rewritten as $\max\{x^\top A y \,|\, \|x\|_2 = \|y\|_2 = 1, \|x\|_0 \le k, \|y\|_0 \le k\}$ — we repeat this algorithm with multiple random initializations. We repeat 20 times and take the best solution. We emphasize that Algorithm 2 may not lead to a global solution of SPCA.

4.3 Implementation of Convex-IP model and Pert-Convex-IP model

4.3.1 Deciding λ, N

1. Deciding λ: The size of the set $\{i : \lambda_i > \lambda\}$ denoted by I_{pos} plays an important role for the computational tractability of our method. So our algorithm inputs an initial value, $I^{\text{ini}}_{\text{pos}}$. From the primal heuristic, we obtain a lower bound $\text{LB}^{\text{primal}}$ on $\lambda^k(A)$. Let

$$\lambda_{i_1} \ge \lambda_{i_2} \ge \cdots \ge \lambda_{i_n},$$

be the eigenvalues of A. If $\lambda_{i_{I^{\text{ini}}_{\text{pos}}}} < \text{LB}^{\text{primal}}$, then we set $\lambda \triangleq \lambda_{i_{I^{\text{ini}}_{\text{pos}}}}$. On the other hand, if $\lambda_{i_{I^{\text{ini}}_{\text{pos}}}} > \text{LB}^{\text{primal}}$, then let l be the smallest index such that $\lambda_{i_l} > \text{LB}^{\text{primal}}$ and we set $\lambda \triangleq \lambda_{i_l}$.

2. Deciding N: In practice, θ_i was found to be significantly smaller than 1. So we used a value of $N = 3$ in all our experiments.

4.3.2 Final details

A total time of 7200 seconds were given to each instance for running the convex IP (any extra time reported in the tables is due to running time of singular value decomposition and primal heuristics). We have run all our experiments with $k = 10, 20$. For the Convex-IP method, we use: $(I^{\text{ini}}_{\text{pos}}, N) = (10, 3)$. For the Pert-Convex-IP method, we let "iter" denote the maximum number of iterations. We used three settings in our experiments:

$$(I^{\text{ini}}_{\text{pos}}, N, \text{iter}) \in \{(5, 3, 10),\ (10, 3, 3),\ (15, 3, 2)\}\,.$$

The overall algorithm using the Pert-Convex-IP model and the Convex-IP model is presented in Appendix E.

4.4 Analysis of dual bound and optimality gap

The approach we take with the Convex-IP and Pert-Convex-IP models involves two kinds of relaxations: (i) the ℓ_1 relaxation (ii) the relaxation of the objective function with piecewise linear functions.

In order to study the tightness of the dual bounds available from these methods, it is desirable to estimate the optimal solution of both the SPCA problem and the ℓ_1-relax problem. We compute a good feasible solution of ℓ_1-relax problem $\max_{\|x\|_2 \le 1, \|x\|_1 \le \sqrt{k}} x^\top A x$ via a heuristic method similar to Algorithm 2. Details are presented in Appendix F. We also find a dual bound to ℓ_1-relax (note that the cutting-planes we added to Pert-Convex-IP, and the inequalities $-\theta_i \le g_i \le \theta_i$ are not valid for ℓ_1 relaxation) by solving an appropriate variant of Pert-Convex-IP (θ_i's are computed so that they are valid for the ℓ_1 problem and no cutting-planes are added).

We conducted these experiments for all instances except the Pitprop data set, since these instances are solved by SDP exactly.

4.5 Data Sets

We do numerical experiments on two types of data sets:

- **Artificial data set:** Tables 1, 2, 3, 4, 5, 6 show examples for artificial/synthetic datasets that are generated from various distributions.
- **Real data set:** Tables 7, 8, 9 show results for real data sets.

Details of these two types of data sets are presented in Appendix G.

4.6 Description of the rows/columns in the tables

Note that the labels for each of the columns in Tables 1, 2, 3, 4, 5, 6, 7, 8, 9 are as follows:

- **Case:** The first part is a name. '**Case 1**' or '**Case 2**' denotes the random instance number. The second part is in the format $(\text{size}, \text{cardinality})$ denoting the number of columns/rows of the A matrix and the right-hand-side of the ℓ_0 constraint of the original SPCA problem.
- **LB-ℓ_0:** denotes the lower bound on the SPCA problem obtained from the (heuristic) Algorithm 2 in Section 4.2.
- **LB-ℓ_1:** denotes the lower bound on the ℓ_1-relax problem obtained from the heuristic method presented in Appendix F.
- **Convex-IP-ℓ_0, Pert-Convex-IP-ℓ_0, Pert-Convex-IP-ℓ_1:** denote the convex integer program which is a relaxation of SPCA problem, the Pert-Convex-IP-ℓ_0 is a relaxation of SPCA problem, the Pert-Convex-IP-ℓ_1 is a relaxation of ℓ_1-relax problem respectively. Details are presented in Section E.
- **SDP:** denotes the semidefinite programming relaxation.
- **UB:** denotes the upper bound obtained from current dual bound method (i.e., Convex-IP-ℓ_0, Pert-Convex-IP-ℓ_0, Pert-Convex-IP-ℓ_1, SDP).
- **gap:** denotes the approximation ratio (duality gap) obtained by the formula $\textbf{gap} = \frac{\text{UB}-\text{LB-}\ell_0}{\text{LB-}\ell_0}$ or $\textbf{gap} = \frac{\text{UB}-\text{LB-}\ell_1}{\text{LB-}\ell_1}$.
- **time:** denotes the total running time—we consider the overall running time due to singular value decomposition, heuristic methods to obtain primal solutions, and solvers (Gurobi, Mosek) used to solve integer programming (set to terminate within 7200 seconds).

The three rows corresponding to Pert-Convex-IP, corresponds to experiments with three settings: $(I_{\text{pos}}, N, \text{iter}) = \{(5, 3, 10),\ (10, 3, 3),\ (15, 3, 2)\}$.

4.7 Conclusions and summary of numerical experiments

Based on numerical results reported in Tables 1, 2, 3, 4, 5, 6, 7, 8, 9 we draw some preliminary observations:

1. **Size of instances solved:**
   - SDP: Because of limitation of hardware and software, the SDP relaxation method does not solve instances with input matrix of size greater than or equal to 300×300.
   - Convex-IP: The convex IP shows better scalability than the SDP relaxation and produces dual bounds for instances with input matrix of size up to 500×500.
   - Pert-Convex-IP: The perturbed convex IP scales significantly better that the other methods. While we experimented with instances up to size 2000×2000, we believe this method will easily scale to larger instances, when $k = 10, 20$ with (I_{pos}, N) being chosen appropriately.

2. **Quality of dual bound:**

   - SDP vs Best of {Convex-IP, Pert-Convex-IP}: While on some instances SDP obtained better dual bounds, this was not the case for all instances. For example, on the 'controlling sparsity' random instances and both the real data sets Eisen-1 and Eisen-2, SDP bounds are weaker.
   - Convex-IP vs Pert-Convex-IP: If the convex IP solved within the time limit, then usually the bound is better than that obtained for Pert-Convex-IP. Note that in some instances (for example, 'Synthetic example' instances in Table 3), the convex IP instance is difficult to solve—but we have a good dual bound when the algorithm terminates (at the time-limit). In such cases, Pert-Convex-IP performs better as it is easy to solve and usually solves within 1 hour. (Similarly, in theory, the upper bounds obtained by Pert-Convex-IP-ℓ_0 should be smaller than Pert-Convex-IP-ℓ_1, if we solve to optimality. But in some cases, Pert-Convex-IP-ℓ_1 may run fast than Pert-Convex-IP-ℓ_0, and achieve better upper bounds.)
   - Overall gaps for Best of {Convex-IP, Pert-Convex-IP}: Except for the random instances of type 'controlling sparsity' of size 1000×1000, and Lymphoma data set, in all other instances at least one method had a gap less that 10%.
   - Cardinality 10 vs Cardinality 20: When the cardinality budget is allowed to increase, based on our numerical results, we can see that the running time of our Convex-IP and Pert-Convex-IP methods do not change a lot, since the parameter of cardinality k of Convex-IP and Pert-Convex-IP method only influences the linear constraint $\sum_{i=1}^{n} y_i \leq \sqrt{k}$, which is more robust to changes in the value of the cardinality k.

3. In some of the cases, the gap obtained by Pert-Convex-IP-ℓ_0 can be explained on the basis of ℓ_1-relaxation. (Compare for example the upper bound from Pert-Convex-IP-ℓ_0 to the lower bound on ℓ_1 relaxation for Reddit (internet) data.) However, in other cases, we conjecture employing finer discretization (more splitting points) of the objective function may yield better dual bounds (for example, 'controlling sparsity' of size 1000×1000, and Lymphoma data set).

Table 1: Spiked Covariance Recovery - Cardinality 10

Case	**LB-ℓ_0**	**Convex-IP-ℓ_0**			**Pert-Convex-IP-ℓ_0**			**SDP**			**LB-ℓ_1**	**Pert-Convex-IP-ℓ_1**		
		UB	gap	Time	UB	gap	Time	UB	gap	Time		UB	gap	Time
Case 1 (200, 10)	511.95	511.98	0.005 %	380	511.99	0.007 %	76	511.959	0.001 %	1277	511.95	512.00	0.010 %	71
					511.98	0.005 %	230					511.99	0.008 %	327
					511.98	0.005 %	1605					511.99	0.008 %	2110
Case 2 (200, 10)	592.45	592.47	0.003 %	469	592.49	0.006 %	615	592.463	0.002 %	1458	592.46	592.50	0.007 %	299
					592.49	0.006 %	236					592.49	0.005 %	305
					592.48	0.005 %	325					592.49	0.005 %	760
Case 1 (300, 10)	414.04	414.15	0.027 %	1692	414.17	0.03 %	642	NaN	-	-	414.09	414.22	0.03 %	540
					414.16	0.029 %	407					414.20	0.026 %	539
					414.15	0.027 %	796					414.19	0.024 %	1720
Case 2 (300, 10)	568.56	568.62	0.011 %	1067	568.65	0.016 %	82	NaN	-	-	568.58	568.68	0.018 %	134
					568.64	0.014 %	493					568.66	0.014 %	898
					568.63	0.012 %	942					568.65	0.012 %	2738
Case 1 (400, 10)	478.24	478.36	0.025 %	2598	478.41	0.04 %	793	NaN	-	-	478.27	478.43	0.033 %	671
					478.39	0.03%	610					478.41	0.029 %	884
					478.38	0.03%	1495					478.41	0.029 %	2045
Case 2 (400, 10)	426.91	427.07	0.037 %	3374	427.15	0.06 %	181	NaN	-	-	426.93	427.16	0.054 %	219
					427.12	0.05 %	846					427.14	0.049 %	683
					427.10	0.04 %	2137					427.11	0.042 %	5352
Case 1 (500, 10)	256.82	257.24	0.164 %	7525	257.37	0.21 %	1345	NaN	-	-	256.84	257.38	0.21 %	1036
					257.29	0.18 %	1512					257.30	0.18 %	1838
					257.25	0.17 %	3279					257.26	0.16 %	5769
Case 2 (500, 10)	551.74	551.90	0.029 %	7196	551.97	0.04 %	152	NaN	-	-	551.78	552.00	0.040 %	318
					551.95	0.04 %	725					551.98	0.036 %	1077
					551.93	0.03 %	1694					551.96	0.033 %	2945
Case 1 (1000, 10)	315.16	NaN	-	-	317.00	0.57 %	1147	NaN	-	-	315.26	317	0.55 %	5811
					316.86	0.52 %	776					316.92	0.53 %	5978
					316.87	0.53 %	3633					316.96	0.54 %	7519
Case 2 (1000, 10)	383.44	NaN	-	-	384.73	0.34 %	2745	NaN	-	-	383.46	384.75	0.34 %	4801
					384.66	0.32 %	403					384.74	0.33 %	7519
					384.76	0.34 %	3643					384.87	0.36 %	7539

Table 2: Spiked Covariance Recovery - Cardinality 20

Case	**LB-ℓ_0**	**Convex-IP-ℓ_0**			**Pert-Convex-IP-ℓ_0**			**SDP**			**LB-ℓ_1**	**Pert-Convex-IP-ℓ_1**		
		UB	gap	Time	UB	gap	Time	UB	gap	Time		UB	gap	Time
Case 1 (200, 20)	516	526.56	2.05 %	493	516.79	0.15 %	67		%		520	520.46	0.09 %	83
					517.13	0.22 %	234					520.56	0.11%	1804
					518.41	0.47 %	3840					521.57	0.30 %	7204
Case 2 (200, 20)	593	598.84	0.98 %	1847	593.68	0.11 %	290		%		596	596.35	0.06 %	806
					593.69	0.12 %	207					596.35	0.06 %	5056
					593.99	0.17 %	1570					596.80	0.13 %	8251
Case 1 (300, 20)	499	502.48	0.70 %	1848	500.01	0.20 %	1097		%		500	500.34	0.07 %	459
					500.02	0.20 %	440					500.38	0.08 %	452
					500.22	0.24 %	1675					500.63	0.13 %	2236
Case 2 (300, 20)	600	606.76	1.13 %	1771	600.63	0.11 %	3959		%		604	604.66	0.11 %	118
					600.63	0.11 %	359					604.38	0.06 %	1166
					600.79	0.13 %	2020					605.00	0.17 %	7209
Case 1 (400, 20)	483	496.24	2.74 %	6398	484.16	0.24 %	449		%		489	489.27	0.06 %	217
					484.64	0.34 %	646					489.55	0.11 %	813
					486.23	0.67 %	2174					490.88	0.38 %	7221
Case 2 (400, 20)	428	436.20	1.92 %	7426	428.47	0.11 %	282		%		434	434.65	0.15 %	187
					428.50	0.12 %	580					434.68	0.16 %	700
					429.07	0.25 %	5791					435.01	0.23 %	7218
Case 1 (500, 20)	294	297.51	1.19 %	7027	294.82	0.28 %	828		%		294	294.98	0.33 %	285
					294.83	0.28 %	729					295.03	0.35 %	627
					295.20	0.41 %	1695					295.45	0.49 %	1696
Case 2 (500, 20)	571	582.18	1.96 %	4628	571.37	0.06 %	373		%		576	576.36	0.06 %	196
					571.48	0.08 %	299					576.51	0.06 %	1195
					572.28	0.22 %	1641					577.53	0.27 %	7225
Case 1 (1000, 20)	414	-	- %	-	416.20	0.53 %	3133		%		415	416.53	0.37 %	4040
					416.09	0.50 %	2760					416.43	0.34 %	3786
					416.06	0.50 %	5844					416.44	0.35 %	6952
Case 2 (1000, 20)	391	-	- %	-	393.02	0.52 %	1054		%		398	399.21	0.30 %	1172
					393.36	0.60 %	1939					399.51	0.38 %	2000
					394.63	0.93 %	5216					400.74	0.69 %	7334

Table 3: Synthetic Example - Cardinality 10

Case	**LB-ℓ_0**	**Convex-IP-ℓ_0**			**Pert-Convex-IP-ℓ_0**			**SDP**			**LB-ℓ_1**	**Pert-Convex-IP-ℓ_1**		
		UB	gap	Time	UB	gap	Time	UB	gap	Time		UB	gap	Time
Case 1 (200, 10)	5634.14	6227.54	10.5 %	6000	5642.24	0.14 %	38	5639.86	0.10 %	1092	5639	5645	0.11 %	99
					5642.66	0.15 %	16					5644	0.09 %	37
					5642.48	0.15 %	186					5643	0.07 %	142
Case 2 (200, 10)	7321.22	7961.8	8.75 %	6000	7330.92	0.13 %	23	7327.84	0.09 %	1086	7327	7333	0.08 %	125
					7330.54	0.13 %	13					7332	0.07 %	35
					7330.30	0.12 %	47					7332	0.07 %	44
Case 1 (300, 10)	4157.46	4423.74	6.40 %	6000	4168.86	0.27 %	83	NaN	-	-	4162	4171	0.22 %	107
					4169.68	0.29 %	21					4170	0.19 %	48
					4168.82	0.27 %	486					4169	0.16 %	474
Case 2 (300, 10)	5135.48	5144.96	0.18 %	6000	5147.40	0.23 %	62	NaN	-	-	5141	5150	0.18 %	163
					5146.94	0.22 %	59					5148	0.14 %	111
					5147.30	0.23 %	58					5147	0.12 %	131
Case 1 (400, 10)	6519.36	6608.10	1.36 %	4762	6533.74	0.22 %	98	NaN	-	-	6526	6538	0.18 %	220
					6534.46	0.23 %	23					6536	0.15 %	74
					6533.82	0.22 %	349					6536	0.15 %	596
Case 2 (400, 10)	5942.04	6003.90	1.04 %	4628	5963.72	0.36 %	56	NaN	-	-	5965	5975	0.17 %	357
					5967.24	0.42 %	29					5974	0.15 %	238
					5966.58	0.41 %	364					5973	0.13 %	331
Case 1 (500, 10)	5125.84	5227.30	1.98 %	6000	5145.36	0.38 %	149	NaN	-	-	5133	5148	0.29 %	341
					5145.42	0.38 %	44					5146	0.25 %	169
					5144.94	0.37 %	132					5145	0.23 %	229
Case 2 (500, 10)	5545.84	5617.78	1.30 %	6000	5567.36	0.39 %	50	NaN	-	-	5560	5573	0.23 %	310
					5567.16	0.38 %	30					5572	0.22 %	241
					5566.68	0.38 %	231					5571	0.20 %	827
Case 1 (1000, 10)	5116.08	NaN	-	-	5145.83	0.58 %	257	NaN	-	-	5124	5149	0.49 %	1037
					5145.44	0.57 %	128					5147	0.45 %	717
					5145.17	0.57 %	1373					5146	0.43 %	3043
Case 2 (1000, 10)	6946.12	NaN	-	-	6973.33	0.39 %	323	NaN	-	-	6947	6974	0.39 %	1017
					6971.17	0.36 %	129					6971	0.35 %	442
					6969.41	0.34 %	1167					6969	0.32 %	1338

Table 4: Synthetic Example - Cardinality 20

Case	**LB-ℓ_0**	**Convex-IP-ℓ_0**			**Pert-Convex-IP-ℓ_0**			**SDP**			**LB-ℓ_1**	**Pert-Convex-IP-ℓ_1**		
		UB	gap	Time	UB	gap	Time	UB	gap	Time		UB	gap	Time
Case 1 (200, 20)	11222	11278	0.50 %	271	11227	0.04 %	66		%		11235	11240	0.04 %	69
					11226	0.04 %	21					11239	0.04 %	12
					11226	0.04 %	78					11238	0.03 %	25
Case 2 (200, 20)	14588	14643	0.38 %	236	14593	0.03 %	90		%		14605	14610	0.03 %	77
					14593	0.03 %	21					14609	0.03 %	13
					14593	0.03 %	25					14609	0.03 %	13
Case 1 (300, 20)	8282	8406	1.50 %	663	8290	0.10 %	128		%		8293	8300	0.08 %	101
					8292	0.12 %	38					8300	0.08 %	25
					8295	0.16 %	374					8300	0.08 %	297
Case 2 (300, 20)	10233	10264	0.30 %	676	10242	0.09 %	127		%		10245	10253	0.08 %	123
					10240	0.07 %	99					10252	0.07 %	48
					10240	0.07 %	310					10251	0.06 %	63
Case 1 (400, 20)	12976	13166	1.46 %	1680	12987	0.08 %	166		%		12998	13008	0.08 %	161
					12986	0.08 %	66					13007	0.07 %	38
					12988	0.09 %	289					13007	0.07 %	153
Case 2 (400, 20)	11809	11945	1.15 %	2636	11819	0.08 %	199		%		11851	11860	0.08 %	257
					11819	0.08 %	87					11859	0.07 %	60
					11818	0.08 %	485					11859	0.07 %	143
Case 1 (500, 20)	10218	10419	1.97 %	3349	10231	0.13 %	314		%		10232	10246	0.14 %	204
					10231	0.13 %	132					10244	0.12 %	74
					10232	0.14 %	202					10243	0.11 %	139
Case 2 (500, 20)	11032	11188	1.41 %	3646	11045	0.12 %	265		%		11059	11072	0.12 %	243
					11044	0.11 %	124					11070	0.10 %	108
					11046	0.13 %	653					11070	0.10 %	827
Case 1 (1000, 20)	10193	-	- %	-	10219	0.26 %	735		%		10210	10234	0.22 %	839
					10217	0.24 %	483					10232	0.20 %	352
					10217	0.24 %	3257					10343	1.20 %	7435
Case 2 (1000, 20)	13867	-	- %	-	13894	0.19 %	932		%		13873	13898	0.18 %	1117
					13892	0.18 %	421					13896	0.17 %	268
					13892	0.18 %	8807					13894	0.15 %	1480

Table 5: Controlling Sparsity - Cardinality 10

Case	LB-ℓ_0	Convex-IP-ℓ_0			Pert-Convex-IP-ℓ_0			SDP			LB-ℓ_1	Pert-Convex-IP-ℓ_1		
		UB	gap	Time	UB	gap	Time	UB	gap	Time		UB	gap	Time
Case 1 (200, 10)	706	707	0.14 %	925	727	2.9 %	117	709	0.42 %	1360	709	729	2.8 %	693
					725	2.6 %	340					727	2.5 %	1193
					725	2.6 %	3663					725	2.3 %	7210
Case 2 (200, 10)	680	681	0.14 %	1195	704	3.53 %	176	688	1.2 %	1148	688	709	3.1 %	566
					703	3.38 %	372					707	2.8 %	991
					704	3.53 %	3672					705	2.5 %	7211
Case 1 (300, 10)	972	986	1.4 %	1958	1010	3.91 %	135	NaN	-	-	983	1016	3.4 %	934
					1009	3.81 %	453					1012	3.0 %	1160
					1008	3.70 %	3635					1009	2.6 %	5744
Case 2 (300, 10)	976	987	1.1 %	3007	1013	3.79 %	278	NaN	-	-	982	1017	3.6 %	1390
					1010	3.48 %	1558					1014	3.3 %	7226
					1012	3.69 %	3772					1013	3.2 %	7229
Case 1 (400, 10)	1239	1255	1.3 %	7207	1288	4.21 %	769	NaN	-	-	1243	1290	3.8 %	835
					1285	3.96 %	699					1286	3.5 %	6548
					1285	3.96 %	3699					1286	3.5 %	7255
Case 2 (400, 10)	1207	1226	1.6 %	7206	1250	3.56 %	221	NaN	-	-	1213	1254	3.4 %	2127
					1249	3.48%	1894					1251	3.1 %	2601
					1248	3.40 %	3697					1250	3.0 %	7243
Case 1 (500, 10)	1498	1529	2.1 %	12180	1576	5.21 %	1026	NaN	-	-	1512	1581	4.6 %	892
					1569	4.74 %	2881					1574	4.1 %	698
					1570	4.81 %	3661					1570	3.8 %	1383
Case 2 (500, 10)	1498	1530	2.1 %	13917	1560	4.14 %	251	NaN	-	-	1507	1565	3.8 %	482
					1559	4.07 %	1039					1561	3.6 %	1323
					1558	4.01 %	3783					1558	3.4 %	7290
Case 1 (1000, 10)	3948	NaN	-	-	6305	59.7 %	2206	NaN	-	-	4009	6344	58 %	4600
					6052	53.3 %	8318					6071	51 %	7492
					5902	49.5 %	3600					5989	49 %	7492
Case 2 (1000, 10)	4002	NaN	-	-	6325	58.1 %	3270	NaN	-	-	4029	6353	58 %	4658
					6040	51.0 %	8356					6013	49 %	7640
					5902	47.6 %	3600					5858	45 %	7581

Table 6: Controlling Sparsity - Cardinality 20

Case	LB-ℓ_0	Convex-IP-ℓ_0			Pert-Convex-IP-ℓ_0			SDP			LB-ℓ_1	Pert-Convex-IP-ℓ_1		
		UB	gap	Time	UB	gap	Time	UB	gap	Time		UB	gap	Time
Case 1 (200, 20)	1341	1354	0.97 %	277	1341.6	0.04 %	154		%		1345	1347	0.15 %	355
					1341.6	0.04 %	1288					1347	0.15 %	210
					1342.0	0.07 %	7220					1347	0.15 %	4185
Case 2 (200, 20)	1287	1308	1.63 %	332	1287.6	0.04 %	121		%		1293	1294	0.08 %	307
					1287.6	0.04 %	1098					1295	0.15 %	145
					1288.1	0.09 %	7220					1294	0.08 %	4973
Case 1 (300, 20)	1839	1862	1.25 %	1019	1849.0	0.54 %	252		%		1853	1861	0.43 %	922
					1848.8	0.53 %	925					1860	0.38 %	1172
					1850.6	0.63 %	7995					1860	0.38 %	7218
Case 2 (300, 20)	1849	1874	1.35 %	707	1852.5	0.19 %	897		%		1860	1867	0.38 %	460
					1854.0	0.27 %	692					1867	0.38 %	642
					1853.9	0.27 %	7230					1867	0.38 %	7218
Case 1 (400, 20)	2339	2373	1.45 %	907	2388.7	2.12 %	287		%		2367	2395	1.18 %	370
					2386.6	2.04 %	5188					2392	1.06 %	1902
					2387.9	2.09 %	7250					2391	1.01 %	7221
Case 2 (400, 20)	2301	2327	%	3106	2355.0	2.35 %	452		%		2317	2360	1.86 %	2092
					2348.9	2.08 %	5164					2353	1.55 %	3388
					2348.3	2.08 %	7250					2350	1.42 %	7224
Case 1 (500, 20)	2858	2925	2.34 %	2773	2966.8	3.81 %	725		%		2871	2976	3.65 %	1590
					2961.6	3.62 %	7270					2969	3.41 %	3277
					2962.5	3.66 %	7270					2967	3.34 %	7229
Case 2 (500, 20)	2832	2899	2.37 %	3015	2931.9	3.53 %	455		%		2869	2935	2.30 %	828
					2928.7	3.41 %	4657					2929	2.09 %	2557
					2928.8	3.42 %	7270					2927	2.02 %	7236
Case 1 (1000, 20)	7535	-	- %	-	9870	31.0 %	1907		%		7649	9963	30.3 %	7374
					9642	28.0 %	5592					9656	26.2 %	7373
					-	- %	-					9781	27.9 %	7383
Case 2 (1000, 20)	7759	-	- %	-	10037	29.4 %	4009		%		7803	10080	29.2 %	6701
					9717	25.2 %	1844					9826	25.9 %	7386
					-	- %	-					9603	23.1 %	7375

Table 7: First six sparse principal components of Pitprops

Cardinality	**LB-ℓ_0**	**Convex-IP-ℓ_0**			**Pert-Convex-IP-ℓ_0**			**SDP**		
		UB	gap	Time	UB	gap	Time	UB	gap	Time
Cardinality 5		3.517	3.2 %	0.40	3.611	6.0 %	0.34	3.458	1.5 %	3.70
Cardinality 2	1.882	1.909	1.4 %	0.23	1.949	3.6 %	0.34	1.882	0 %	2.49
Cardinality 2	1.364	1.417	3.8 %	0.30	1.468	7.6 %	0.85	1.377	1.0 %	2.69
Cardinality 1	1	1.018	1.8 %	0.75	1.035	3.5 %	1.02	1	0 %	2.40
Cardinality 1	1	1.022	2.2 %	0.30	1.036	3.6 %	0.61	1	0 %	2.42
Cardinality 1	1	1.012	1.2 %	0.30	1.021	2.1 %	0.51	1	0 %	2.32
Sum of above	9.652	9.897	2.5 %	2.28	10.12	4.8 %	3.67	9.717	0.7 %	16.02

Table 8: Biological and Internet Data - Cardinality 10

Case	**LB-ℓ_0**	**Convex-IP-ℓ_0**			**Pert-Convex-IP-ℓ_0**			**SDP**			**LB-ℓ_1**	**Pert-Convex-IP-ℓ_1**		
		UB	gap	Time	UB	gap	Time	UB	gap	Time		UB	gap	Time
Eisen-1 (79, 10)	17.33	17.39	0.3 %	4.6	17.35	0.12 %	63	17.71	2.2 %	15	17.70	17.71	0.06 %	519
					17.36	0.17 %	113					17.71	0.06 %	339
					17.40	0.4 %	412					17.71	0.06 %	2092
Eisen-2 (118, 10)	11.71	11.87	1.4 %	96	12.19	4.10 %	69	11.94	2.0 %	52	11.94	12.38	3.7 %	411
					11.96	2.13 %	139					12.08	1.2 %	352
					11.91	1.70 %	385					12.02	0.7 %	1189
Colon (500, 10)	2641	3028	14.7 %	9000	3373	27.7 %	708	NaN	-	-	2759	3456	25 %	2024
					2894	9.58 %	1181					2965	7.5 %	4108
					2823	6.89 %	353					2900	5.1 %	7273
Lymphoma (500, 10)	6008	7583	20.7 %	3723	8470	41 %	610	NaN	-	-	6289	8730	39 %	1439
					7278	21 %	1526					7469	19 %	2637
					7031	17 %	2808					7190	14 %	7278
Reddit (2000, 10)	1523	NaN	-	-	1712	% 12.41	5932	NaN	-	-	1733	1861	7.4 %	8674
					1665	% 9.32	8681					1808	4.3 %	8718
					1693	% 11.16	8536					1839	6.1 %	8867

5 Acknowledgements

We would like to thank Munmun De Choudhury for providing us with the internet data set.

References

[1] Genevera I Allen and Mirjana Maletić-Savatić. Sparse non-negative generalized pca with applications to metabolomics. *Bioinformatics*, 27(21):3029–3035, 2011.

[2] Shrey Bagroy, Ponnurangam Kumaraguru, and Munmun De Choudhury. A social media based index of mental well-being in college campuses. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1634–1646, New York, NY, USA, 2017. ACM.

[3] Lauren Berk and Dimitris Bertsimas. Certifiably optimal sparse principal component analysis. *technical report*, 2016.

[4] Quentin Berthet and Philippe Rigollet. Optimal detection of sparse principal components in high dimension. *The Annals of Statistics*, 41(4):1780–1815, 2013.

Table 9: Biological and Internet Data - Cardinality 20

Case	LB-ℓ_0	Convex-IP-ℓ_0			Pert-Convex-IP-ℓ_0			SDP			LB-ℓ_1	Pert-Convex-IP-ℓ_1		
		UB	gap	Time	UB	gap	Time	UB	gap	Time		UB	gap	Time
Eisen-1 (79, 20)	17.71	17.94	1.30 %	742	17.73	0.11 %	757	18.13	2.37%	13	18.13	18.25	0.66 %	4970
					17.73	0.11 %	1023					18.13	0.00 %	1732
					17.74	0.17 %	7204					18.19	0.33 %	7200
Eisen-2 (118, 20)	19.32	19.71	2.02 %	64	19.57	1.29 %	200	19.76	2.28%	53	19.76	19.97	1.06 %	289
					19.42	0.52 %	149					19.83	0.35 %	234
					19.46	0.72 %	236					19.81	0.25 %	757
Colon (500, 20)	4255	4907	15.3 %	7230	4958	16.5 %	1625	NaN	- %	-	4531	5186	14.5 %	2778
					4500	5.76 %	7285					4733	4.46 %	5883
					4433	4.18 %	7286					4721	4.19 %	7230
Lymphoma (500, 20)	9082	10783	18.7 %	7239	11122	22.5 %	871	NaN	- %	-	9701	11533	18.9 %	1686
					10157	11.8 %	2798					10929	12.7 %	7250
					10020	10.3 %	7293					10434	7.56 %	7254
Reddit (2000, 20)	1119	-	- %	-	1151	2.86 %	4715	NaN	- %	-	1223	1235	0.98 %	7756
					1150	2.77 %	8628					1235	0.98 %	7742
					-	- %	-					1282	4.82 %	7731

[5] Daniel Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical programming*, 74(2):121–140, 1996.

[6] Pierre Bonami, Oktay Günlük, and Jeff Linderoth. Solving box-constrained nonconvex quadratic programs. *Optimization online*, pages 26–76, 2016.

[7] Samuel Burer and Dieter Vandenbussche. Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound. *Computational Optimization and Applications*, 43(2):181–195, 2009.

[8] Jorge Cadima and Ian T Jolliffe. Loading and correlations in the interpretation of principle compenents. *Journal of Applied Statistics*, 22(2):203–214, 1995.

[9] Siu On Chan, Dimitris Papailiopoulos, and Aviad Rubinstein. On the worst-case approximability of sparse pca. *arXiv preprint arXiv:1507.05950*, 2015.

[10] A. d'Aspremont, L. El. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM Review*, 49:434–448, 2007.

[11] Alexandre d'Aspremont, Francis R Bach, and Laurent El Ghaoui. Full regularization path for sparse principal component analysis. In *Proceedings of the 24th international conference on Machine learning*, pages 177–184. ACM, 2007.

[12] Alexandre d'Aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse pca using semidefinite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

[13] Alexandre dAspremont, Francis Bach, and Laurent El Ghaoui. Approximation bounds for sparse principal component analysis. *Mathematical Programming*, 148(1-2):89–110, 2014.

[14] Alexandre dAspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(Jul):1269–1294, 2008.

[15] Antonio Frangioni and Claudio Gentile. Sdp diagonalizations and perspective cuts for a class of nonseparable miqp. *Operations Research Letters*, 35(2):181–185, 2007.

[16] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity.* CRC press, 2015.

[17] Yunlong He, Renato DC Monteiro, and Haesun Park. An algorithm for sparse pca based on a new sparsity control criterion. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 771–782. SIAM, 2011.

[18] JNR Jeffers. Two case studies in the application of principal component analysis. *Applied Statistics*, pages 225–236, 1967.

[19] Ian T Jolliffe, Nickolay T Trendafilov, and Mudassir Uddin. A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics*, 12(3):531–547, 2003.

[20] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(Feb):517–553, 2010.

[21] Jinhak Kim. *Cardinality Constrained Optimization Problems.* PhD thesis, Purdue University, West Lafayette, Indiana, 8 2016.

[22] Shiqian Ma. Alternating direction method of multipliers for sparse principal component analysis. *Journal of the Operations Research Society of China*, 1(2):253–274, Jun 2013.

[23] Malik Magdon-Ismail. Np-hardness and inapproximability of sparse pca. *Information Processing Letters*, 126:35–38, 2017.

[24] George L Nemhauser and Laurence A Wolsey. *Integer and Combinatorial Optimization. Interscience Series in Discrete Mathematics and Optimization.* 1988.

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

[26] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[27] Koustuv Saha and Munmun De Choudhury. Modeling stress with social media around incidents of gun violence on college campuses. *Proc. ACM Hum.-Comput. Interact.*, 1(CSCW):92:1–92:27, December 2017.

[28] Yla R Tausczik and James W Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54, 2010.

[29] Roman Vershynin. *High-Dimensional Probability An Introduction with Applications in Data Science.* Draft, 2016.

[30] Tengyao Wang, Quentin Berthet, and Richard J Samworth. Statistical and computational trade-offs in estimation of sparse principal components. *The Annals of Statistics*, 44(5):1896–1930, 2016.

[31] DM. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[32] Adams Wei Yu, Hao Su, and Li Fei-Fei. Efficient euclidean projections onto the intersection of norm balls. *arXiv preprint arXiv:1206.4638*, 2012.

[33] Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, 14(Apr):899–925, 2013.

[34] Youwei Zhang, Alexandre dAspremont, and Laurent El Ghaoui. Sparse pca: Convex relaxations, algorithms and applications. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 915–940. Springer, 2012.

[35] Zhenyue Zhang, Hongyuan Zha, and Horst Simon. Low-rank approximations with sparse factors i: Basic algorithms and error analysis. *SIAM Journal on Matrix Analysis and Applications*, 23(3):706–727, 2002.

[36] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

A SDP relaxation

The SPCA problem $\max_{\|x\|_2=1,\|x\|_0\leq k} x^\top Ax$ is equivalent to a nonconvex problem:

$$\begin{aligned}&\max\ \mathrm{tr}(AX)\\ &\text{s.t. } \mathrm{tr}(X)=1, \|X\|_0\leq k^2, X\succeq 0, \mathrm{rank}(X)=1.\end{aligned}$$

Further relaxing this by replacing its rank and cardinality constraints with $\mathbf{1}^\top X\mathbf{1}\leq k$ gives the standard SDP relaxation:

$$\begin{aligned}&\max\ \mathrm{tr}(AX)\\ &\text{s.t. } \mathrm{tr}(X)=1, \mathbf{1}^\top X\mathbf{1}\leq k, X\succeq 0.\end{aligned}\tag{SDP}$$

B Proof of Proposition 1

Proof. **Proof of Proposition 1:** Let $x^*=(x_i^*)_{i=1}^n$ be an optimal solution of SPCA. Then set

$$\begin{cases} g_i^* & \leftarrow & (x^*)^\top v_i, & i\in[n],\\ \left((\eta_i^{-N})^*,\ldots,(\eta_i^N)^*\right) & \leftarrow & \left(\eta_i^{-N},\ldots,\eta_i^N\right)\in \text{SOS-2 and } \sum_{j=-N}^N \gamma_i^j(\eta_i^j)^*=g_i^*, & i\in\{i:\lambda_i>\lambda\},\\ \xi_i^* & \leftarrow & \sum_{j=-N}^N(\gamma_i^j)^2\eta_i^j, & i\in\{i:\lambda_i>\lambda\},\\ y_i^* & \leftarrow & |x_i^*|, & i\in[n],\\ s_i^* & \leftarrow & \sum_{i\in\{i:\lambda_i\leq\lambda\}} -(\lambda_i-\lambda)g_i^*. \end{cases}$$

Note that the above solution $(x^*, y^*, g^*, \xi^*, \eta^*, s^*)$ is a feasible solution for Convex-IP. This is easy to verify for all the constraints except the constraint $\sum_{i\in\{i:\lambda_i>\lambda\}} \xi_i + \sum_{i\in\{i:\lambda_i\leq\lambda\}} g_i^2 \leq 1 + \frac{1}{4N^2}\sum_{i\in\{i:\lambda_i>\lambda\}} \theta_i^2$. Note that to verify this constraint, it is sufficient to verify that $\xi_i \leq g_i^2 + \frac{1}{4N^2}\theta_i^2$ for $i \in \{i : \lambda_i > \lambda\}$. This is easily verified based on the size of the discretization and the structure of SOS-2 constraints.

Moreover, the objective value of feasible solution $(x^*, y^*, g^*, \xi^*, \eta^*, s^*)$ is

$$\begin{aligned}
\lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\xi_i^* - s^* \geq & \lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)(g_i^*)^2 - s^* \\
= & \lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)((x^*)^\top v_i)^2 + \sum_{i\in\{i:\lambda_i\leq\lambda\}} (\lambda_i - \lambda)((x^*)^\top v_i)^2 \\
= & \lambda + \sum_{i=1}^{n} (\lambda_i - \lambda)((x^*)^\top v_i)^2.
\end{aligned}$$

Note that the optimal solution x^* of SPCA has property $\|x^*\|_2 = 1$ and $\sum_{i=1}^n v_i v_i^\top = I_n$. Then $\lambda + \sum_{i=1}^n (\lambda_i - \lambda)((x^*)^\top v_i)^2 = (x^*)^\top A x^* = \lambda^k(A)$. Therefore, $\mathrm{OPT}_{\text{convex-IP}} \geq \lambda^k(A)$. □

C Proof of Proposition 2

Proof. **Proof of Proposition 2:** Let $(\bar{x}, \bar{y}, \bar{g}, \bar{\xi}, \bar{\eta}, \bar{s})$ be an optimal solution for Convex-IP. Its optimal value then satisfies the following:

$$\begin{aligned}
\mathrm{OPT}_{\text{convex-IP}} &= \lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\bar{\xi}_i - \bar{s} \\
&= \lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\left(\bar{\xi}_i - \bar{g}_i^2 + \bar{g}_i^2\right) - \bar{s} \\
&= \lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\left(\bar{\xi}_i - \bar{g}_i^2\right) + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\bar{g}_i^2 - \bar{s}.
\end{aligned}$$

Since variable s satisfies $\sum_{i\in\{i:\lambda_i\leq\lambda\}} -(\lambda_i - \lambda)g_i^2 \leq s$, to maximize the objective function, $\bar{s}$ should be equivalent to $\sum_{i\in\{i:\lambda_i\leq\lambda\}} -(\lambda_i - \lambda)\bar{g}_i^2$, then the above formula can be represented as

$$\begin{aligned}
& \lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\left(\bar{\xi}_i - \bar{g}_i^2\right) + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\bar{g}_i^2 - \bar{s} \\
= & \lambda + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\left(\bar{\xi}_i - \bar{g}_i^2\right) + \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\bar{g}_i^2 + \sum_{i\in\{i:\lambda_i\leq\lambda\}} (\lambda - \lambda)\bar{g}_i^2 \\
= & \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\left(\bar{\xi}_i - \bar{g}_i^2\right) + \left(\lambda + \sum_{i=1}^{n} (\lambda_i - \lambda)\bar{g}_i^2\right). \qquad (18)
\end{aligned}$$

By previous results, $\lambda + \sum_{i=1}^n (\lambda_i - \lambda)\bar{g}_i^2 = \bar{x}^\top A \bar{x}$. Note that due to the ℓ_2-norm constraint $\|x\|_2 \leq 1$ and the ℓ_1-norm constraint present in Convex-IP problem, we have $\bar{x} \in T_k = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1, \|x\|_1 \leq \sqrt{k}\} \subseteq \rho \cdot \mathrm{Conv}\,(S_k)$. Therefore $\bar{x}^\top A \bar{x}$ is upper bounded by the value $\rho^2 \cdot \lambda^k(A)$. For the

first term in (18), since our SOS-2 construction enforces $g_i^2 + \frac{\theta_i^2}{4N^2} \geq \xi_i \geq g_i^2$, we obtain:

$$\begin{aligned}\mathrm{OPT}_{\text{convex-IP}} &= \sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\left(\bar{\xi}_i - \bar{g}_i^2\right) + \left(\lambda + \sum_{i=1}^{n}(\lambda_i - \lambda)\bar{g}_i^2\right) \\ &\leq \frac{1}{4N^2}\sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\theta_i^2 + \rho^2 \cdot \lambda^k(A).\end{aligned}$$

□

D Proof of Proposition 4

Proof. **Proof of Proposition 4:** Based on Proposition 2, we have

$$\mathrm{OPT}_{\text{Pert-Convex-IP}} \leq \rho^2\lambda^k(\bar{A}) + \frac{1}{4N^2}\sum_{i\in\{i:\lambda_i>\lambda\}} (\lambda_i - \lambda)\theta_i^2.$$

Note that $\bar{A} - A = \sum_{i\in\{i:\lambda_i\leq\lambda\}}(\bar{\lambda} - \lambda_i)v_i v_i^\top$. Therefore,

$$\begin{aligned}\rho^2\lambda^k(\bar{A}) &= \rho^2\lambda^k\left(A + (\bar{A} - A)\right) \\ &\leq \rho^2\lambda^k(A) + \rho^2\lambda^k(\bar{A} - A) \\ &\leq \rho^2\lambda^k(A) + \rho^2(\bar{\lambda} - \lambda_{\min}(A)).\end{aligned}$$

□

E Convex-IP Method and Pert-Convex-IP Method

Algorithm 3 presents all the details of the convex IP solved. Algorithm 4 presents all the details of the Pert-Convex-IP solved.

F Primal heuristic for ℓ_1 relaxation

We use a algorithm very similar to the primal algorithm 2 described in Section 4.2.

One key step in this algorithm is the following: Given any vector $v = (v_i)_{i=1}^n \in \mathbb{R}^n$ with ℓ_2-ball $B_{\ell_2,1} \triangleq \{x : \|x\|_2 \leq 1\}$ and ℓ_1-ball $B_{\ell_1,\sqrt{k}} \triangleq \{x : \|x\|_1 \leq \sqrt{k}\}$, we want to find out the vector u as a solution of $\min_{u\in B_{\ell_2,1}\cap B_{\ell_1,\sqrt{k}}} \|v - u\|_2$. First, without loss of generality, we may assume that $v_i \geq 0$ holds for $i = 1, \ldots, n$. Since for a fixed v, let $u^* \leftarrow \arg\min_{u\in B_{\ell_2,1}\cap B_{\ell_1,\sqrt{k}}} \|v - u\|_2$, we claim that $\mathrm{sign}(u_i^*) = \mathrm{sign}(v_i)$ for $i = 1, \ldots, n$, otherwise one can change the sign of the components of u^* that do not match and remain the optimality. Second, when we have $v \geq 0$, the constraints of ℓ_1-ball $B_{\ell_1,\sqrt{k}}$ are equivalent to $\sum_{i=1}^n v_i \leq \sqrt{k}$. Therefore, the optimization problem is convex with only two constraints. In paper [32], Hao, Yu and Li gave an efficient way to project any vector in $\mathbb{R}^n$ onto the intersection of norm balls.

In our paper, we solve this convex optimization problem using Gurobi 7.0.2. We used $\epsilon = 10^{-6}$ and $N = 20$.

Algorithm 3 Convex-IP Method

Input: Sample covariance matrix A, cardinality constraint k, size of set $\{i : \lambda_i > \lambda\}$ we desire, number of one branch splitting points N.
Output: Lower and upper bound of SPCA or ℓ_1-relax based on the choice of θ_i.
function CONVEX-IP METHOD$(A, k, I_{\text{pos}}, N)$
    Set lower bound and warm starting point $(\text{LB}, \bar{x}) \leftarrow$ HEURISTIC METHOD(A, k, x^0).
    Set parameter $\lambda_{I_{\text{pos}+1}} \leq \lambda \leq \text{LB}$ if possible, otherwise set $\lambda \leftarrow \text{LB}$.
    Set splitting points γ_i^j as above based on N and the choice of θ_i, see Section 2.2 [3] .
    To warm start, add additional splitting points based on the point $\bar{x}$.
    Add cutting-plane (9) to the model based on the choice of θ_i.
    Run Convex-IP problem.
    Set UB $\leftarrow$ Convex-IP if running to the optimal, or the current dual bound obtained from Convex-IP.
    return LB, UB.
end function

Algorithm 4 Pert-Convex-IP Method

Input: Sample covariance matrix A, cardinality constraint k, size of set $\{i : \lambda_i > \lambda\}$ we desire, number of one branch splitting points N, maximum number of iterations iter.
Output: Lower and upper bound of SPCA or ℓ_1-relax based on the choice of θ_i.
function PERT-CONVEX-IP METHOD$(A, k, I_{\text{pos}}, N, \text{iter})$
    Set lower bound and warm starting point $(\text{LB}, \bar{x}) \leftarrow$ HEURISTIC METHOD(A, k, x^0).
    Set parameter $\lambda_{I_{\text{pos}+1}} \leq \lambda \leq \text{LB}$ if possible, otherwise set $\lambda \leftarrow \text{LB}$.
    Set parameter $\bar{\lambda} \triangleq \max\{\lambda_i : \lambda_i \leq \lambda\} < \lambda$ if possible.
    Set splitting points γ_i^j as above based on N and the choice of θ_i, see Section 2.2 [3].
    To warm start, add additional splitting points based on the point $\bar{x}$.
    while current iteration does not exceed the maximum number of iterations iter or time limit is not up **do**
        Run Pert-Convex-IP problem.
        Set UB $\leftarrow$ Pert-Convex-IP if running to the optimal, or the current dual bound obtained from Pert-Convex-IP.
        Set $\hat{x} \leftarrow$ current feasible solution obtained from Pert-Convex-IP
        Add additional splitting points based on solution obtained in solving Pert-Convex-IP problem.
        Add cutting-plane (9) to the model based on the choice of θ_i.
    end while
    return LB, UB.
end function

G Description of Data Sets

G.1 Artificial Data Sets

We first conduct numerical experiments on three types of artificial data sets, denoted as the spiked covariance recovery from the paper [25], the synthetic example from the paper [36], and the con-

Algorithm 5 Primal ℓ_1-relaxation heuristic

Input: Sample covariance matrix A, cardinality constriant k, initial feasible vector x.
Output: A feasible solution x of ℓ_1-relax and its objective value.
function ℓ_1-RELAXATION HEURISTIC(A, k, x)
    Start from a vector x such that $\|x\|_2 \leq 1, \|x\|_1 \leq \sqrt{k}$.
    Set current solution be $x^{\text{current}} \leftarrow x$, and past solution be $x^{\text{past}} \leftarrow 0$.
    Set current objective value be $\text{OPT}^{\text{current}} \leftarrow (x^{\text{current}})^\top A x^{\text{current}}$, and past objective value be $\text{OPT}^{\text{past}} \leftarrow (x^{\text{past}})^\top A x^{\text{past}}$.
    Let iter denotes the index of current iteration, set N be the maximum number of iterations, set ϵ be a stopping criteria.
    while $\text{OPT}^{\text{current}} > \text{OPT}^{\text{past}}$ and $\|x^{\text{current}} - x^{\text{past}}\|_2 > \epsilon$ and iter $\leq N$ **do**
        Set $y \leftarrow A x^{\text{current}}$.
        Denote the projection of y onto the set $P = \{y : \|y\|_2 \leq 1, \|y\|_1 \leq \sqrt{k}\}$ as $\text{Proj}_P(y)$.
        Set $x^{\text{past}} \leftarrow x^{\text{current}}$, and $x^{\text{current}} \leftarrow \text{Proj}_P(y)$.
        Set $\text{OPT}^{\text{current}} \leftarrow (x^{\text{current}})^\top A x^{\text{current}}$, and $\text{OPT}^{\text{past}} \leftarrow (x^{\text{past}})^\top A x^{\text{past}}$.
    end while
    return $x^{\text{current}}, \text{OPT}^{\text{current}}$.
end function

trolling sparsity case from the paper [12]. A description of each of these three types of instances is presented below:

G.1.1 Spiked covariance recovery

Consider a covariance matrix Σ, which has two sparse eigenvectors with dominated eigenvalues and the rest eigenvector are unconstrained with small eigenvalues. Let the first two dominant eigenvectors v_1, v_2 of Σ be:

$$[v_1]_i = \begin{cases} \frac{1}{\sqrt{10}} & i = 1, \ldots, 10, \\ 0 & \text{otherwise} \end{cases}, \qquad [v_2]_i = \begin{cases} \frac{1}{\sqrt{10}} & i = 11, \ldots, 20, \\ 0 & \text{otherwise} \end{cases}, \tag{19}$$

with the eigenvalues corresponding to the first two dominant eigenvectors be $\lambda_1 \gg 1$ and $\lambda_2 \gg 1$, and the remaining eigenvalues be 1. For example, in our numerical experiments, set $\Sigma \leftarrow 399 \cdot v_1 v_1^\top + 299 \cdot v_2 v_2^\top + I$.

We have four distinct settings under the spiked covariance recovery case. Let n be the number of features, i.e., the size of the sample covariance matrix of our numerical cases. Let m be the number of samples we generated. We set $n = \{200, 300, 400, 500, 1000\}$ and $m = \{50\}$. Therefore, under each setting of n, we generate m random samples $x_i \sim N(0, \Sigma)$, and get our sample covariance matrix $\hat{\Sigma} = \frac{1}{50} \sum_{i=1}^{50} x_i x_i^\top$. In Table 1, for each setting, we repeat the experiment for 2 times (case 1, case 2), and compare the dual bounds obtained from all three methods.

G.1.2 Synthetic Example

Given n, let $n_1, n_2, n_3 \in \left\{\lceil \frac{n}{3} \rceil, \lfloor \frac{n}{3} \rfloor\right\}$ such that $n_1 + n_2 + n_3 = n$. Let $\mathbf{0}_{p\times q}$ be the matrix of all zeros with size $p \times q$. Let $\mathbf{1}_p$ be the vector of all ones with length p. Then:

$$\Sigma = \begin{pmatrix} 290 \cdot \mathbf{1}_{n_1}\mathbf{1}_{n_1}^\top + I_{n_1} & \mathbf{0}_{n_1\times n_2} & -87 \cdot \mathbf{1}_{n_1}\mathbf{1}_{n_3}^\top \\ \mathbf{0}_{n_2\times n_1} & 300 \cdot \mathbf{1}_{n_2}\mathbf{1}_{n_2}^\top + I_{n_2} & 277.5 \cdot \mathbf{1}_{n_2}\mathbf{1}_{n_3}^\top \\ -87 \cdot \mathbf{1}_{n_3}\mathbf{1}_{n_1}^\top & 277.5 \cdot \mathbf{1}_{n_3}\mathbf{1}_{n_2}^\top & 582.7875 \cdot \mathbf{1}_{n_3}\mathbf{1}_{n_3}^\top + I_{n_3} \end{pmatrix}. \tag{20}$$

In our experiments, we set $n = \{200, 300, 400, 500, 1000\}$, and generate $m = 50$ samples such that $x_i \sim N(0, \Sigma)$. Again, the sample empirical covariance matrix is $\hat{\Sigma} = \frac{1}{50}\sum_{i=1}^{50} x_i x_i^\top$. In Table 3, for each setting of n, we repeat the experiment twice (case 1, case 2), and compare dual bounds obtained from all three methods.

G.1.3 Controlling Sparsity

Like the spiked covariance recovery case, the covariance matrix Σ of controlling sparsity case can also be represented as the summation of a term generated by sparse eigenvector with dominated eigenvalue and the remaining part with small eigenvalues. Generate a $n\times n$ matrix U with uniformly distributed coefficients in $[0, 1]$ which can be seen as white noise. Let $v \in \{0, 1\}^n$ be a sparse vector with $\|v\|_0 \leq k$. We then form a test matrix $\Sigma = U^\top U + \sigma vv^\top$, where σ is the signal-to-noise ratio and is set to 15.

In our experiments, we set $n = \{200, 300, 400, 500, 1000\}$ and generate $m = 50$ samples $x_i \sim N(0, \Sigma)$ for $i = 1, \ldots, 50$. Therefore the sample empirical covariance matrix is $\hat{\Sigma} = \frac{1}{50}\sum_{i=1}^{50} x_i x_i^\top$. In Table 5, for each setting of n, we repeat the experiment twice (case 1, case 2), and compare dual bounds obtained from all three methods.

G.2 Real Data Sets

We conduct numerical experiments on three types of real data sets, the benchmark pitprops data from [18], biological data from [33], [11], [25], and large-scale data collected from internet.

G.2.1 Pitprops Data

The PitProps data set in [18] (consisting of 180 observations with 13 measured variables) has been a standard benchmark to evaluate algorithms for sparse PCA.

Based on previous work, we also consider the first six $k-$sparse principal components. Note the i-th $k-$sparse principal component x^i is obtained by solving $\arg\max_{\|x\|_2=1, \|x\|_0\leq k} x^\top A^i x$ where $A^1 \leftarrow A$ and $A^i \leftarrow (I - x^{i-1}(x^{i-1})^\top)A^{i-1}(I - x^{i-1}(x^{i-1})^\top)$ for $i = 2, \ldots, 6$. Table 7 lists the six extracted sparse principal direction with cardinality setting $5 - 2 - 2 - 1 - 1 - 1$.

G.2.2 Biological Data

In Table 8 we present numerical experiments on four biological data sets. The first two biological data sets (Eisen-1, Eisen-2) are from [33]. The Colon cancer data set is from Alon et al. (1999). The Lymphoma data set is from Alizadeh et al. (2000).

G.2.3 Large-scale Internet Data

In Table 8 we also present numerical experiments on internet dataset. This dataset is constructed out of textual posts shared on the popular social media Reddit. Based on prior work [2, 27], the archive of all public Reddit posts shared on Googles Big Query was utilized to obtain a set of 3292 posts from the subreddit r/stress from December 2010 to January 2017. The r/stress community allows individuals to self-report and disclose their stressful experiences and is a support community. For example, two (paraphrased) post excerpts say: "Feel like I am burning out (again...) Help: what do I do?"; and "How do I calm down when I get triggered?". The community is also heavily moderated; hence these 3292 posts were considered to be indicative of actual stress. [27].

Then on this collected set of posts, standard text-based feature extraction techniques were applied per post, starting with cleaning the data (stopword elimination, removal of noisy words, stemming), and then building a language model with the n-grams in a post (n=2). The outcomes of this language model provided us with 1950 features, after including only the top most statistically significant features. Additionally, the psycholinguistic lexicon Linguistic Inquiry and Word Count (LIWC) [26] was leveraged to obtain features aligning with 50 different empirically validated psychological categories, such as positive affect, negative affect, cognition, and function words. These features have been extensively validated in prior work to be indicative of stress and similar psychological constructs [28]. Our final dataset matrix comprised 3092 rows, corresponding to the 3092 posts, and 2000 features in all.

The purpose of testing the sparse PCA technique on this dataset is to identify those features that are theoretically guaranteed to be the most salient in describing the nature of stress expressed in a post. In turn, these salient features could be utilized by a variety of stakeholders like clinical psychologists, and community moderators and managers to gain insights into stress-related phenomenon as well as to direct interventions as appropriate.

The final A matrix can be found on the website:
https://www2.isye.gatech.edu/ sdey30/publications.html