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 $2000 \times 2000$.

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

$$\max_{\|x\|_2=1} x^\top Ax \quad \text{(PCA)}$$
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:

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

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 $O(n^d)$ 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 \( 2000 \times 2000 \).

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 \( \ell_1 \)-norm relaxation of SPCA:

\[
\text{OPT}_{\ell_1} \triangleq \max_{\|x\|_2 \leq 1, \|x\|_1 \leq \sqrt{k}} x^\top Ax. \quad (\ell_1\text{-relax})
\]

The relaxation \( \ell_1\)-relax has two advantages: (a) As shown in Theorem 1 below, \( \ell_1\)-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 \( \ell_1\)-relax (together with some implied linear inequalities for SPCA) which heavily use the fact that the feasible region of \( \ell_1\)-relax is convex.

We note that \( \ell_1\)-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 \( \ell_1 \)-norm (as opposed to a small \( \ell_0 \)-norm). However, we emphasize that \( \ell_1\)-relax is a nonconvex optimization problem—globally optimizing \( \ell_1\)-relax is challenging—we show in this paper, how one can use IP methods to solve \( \ell_1\)-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 \( \ell_1\)-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 \( \ell_1\)-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 \( \text{OPT}_{\ell_1} \) is upper bounded by a multiplicative factor \( \rho^2 \) away from \( \lambda^k(A) \), i.e., \( \lambda^k(A) \leq \text{OPT}_{\ell_1} \leq \rho^2 \cdot \lambda^k(A) \) with \( \rho \leq 1 + \sqrt{k/(k+1)} \).

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 \( \ell_1\)-relax. In particular, the proof of Theorem 1 implies that if one can obtain a solution for \( \ell_1\)-relax which is within a constant factor, say \( \theta \), of \( \text{OPT}_{\ell_1} \),
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 $\ell_1$-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 $\ell_0$-norm constraint, resulting in a very large number of binary variables. In particular, a typical model could be of the form:

$$\begin{align*}
\max & \quad \text{tr}(AX) \\
\text{s.t.} & \quad x_i \leq z_i, \quad i \in [n] \\
& \quad \sum_{j=1}^{n} z_i \leq k \\
& \quad \|x\|_2 \leq 1 \\
& \quad \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \succeq 0 \\
& \quad \text{rank} \left( \begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix} \right) = 1 \\
& \quad z \in \{0,1\}^n.
\end{align*}$$

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 $\ell_1$-relax is a convex set. Therefore, we do not have to include binary variables as in the case for modeling the $\ell_0$-norm constraint. Moreover, Theorem 1 suggests that $\ell_1$-relax will provide quite strong dual bounds for SPCA. Thus, we will use $\ell_1$-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 $\ell_1$-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 $\ell_1$-relax is:

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

2. Assuming that $\lambda \leq \lambda^k(A)$, we have that $x^\top Ax = x^\top (A - I\lambda)x + \lambda$ for $x$ such that $\|x\|_2 = 1$, where $I$ is the identity matrix. Therefore, if we split the eigenvalues into two sets as $\{ i : \lambda_i >$
\[ \lambda \text{ and } \{ i : \lambda_i \leq \lambda \}, \text{ the objective function can be represented as} \]
\[ \lambda + \sum_{i \in \{i: \lambda_i > \lambda \}} (\lambda_i - \lambda)(x^T v)_i^2 + \sum_{i \in \{i: \lambda_i \leq \lambda \}} (\lambda_i - \lambda)(x^T v)_i^2. \]

3. For each index \( i \in \{ i : \lambda_i > \lambda \} \), we replace \( x^T v_i \) with a single continuous variable \( g_i \), and set \( \theta_i \leftarrow \max\{x^T v_i : \|x\|_2 \leq 1, \|x\|_0 \leq k \} \) (or \( \theta_i \leftarrow \max\{x^T v_i : \|x\|_2 \leq 1, \|x\|_1 \leq \sqrt{k} \} \) if we explicitly want a relaxation of \( \ell_1\)-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 \( \xi_i \) 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 \( \sum_{i \in \{i: \lambda_i \leq \lambda \}} (\lambda_i - \lambda)(x^T v)_i^2 \), since \( \lambda_i - \lambda \leq 0 \), we obtain a convex constraint \( \sum_{i \in \{i: \lambda_i \leq \lambda \}} (\lambda_i - \lambda)(x^T v)_i^2 \leq s \).

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

\[
\begin{align*}
\lambda + \max_{x,y,g,\xi,\eta,s} & \sum_{i \in \{i: \lambda_i > \lambda \}} (\lambda_i - \lambda)\xi_i - s \quad (\triangleq \text{OPT}_{\text{Convex-IP}}) \\
g_i = x^T v_i, & \quad i \in [n] \\
g_i = \sum_{j=-N}^{N} \lambda_i^j \eta_i^j, & \quad i \in \{ i : \lambda_i > \lambda \} \\
\xi_i = \sum_{j=-N}^{N} (\lambda_i^j)^2 \eta_i^j, & \quad i \in \{ i : \lambda_i > \lambda \} \\
(\eta_i^{-N}, \ldots, \eta_i^N) & \in \text{SOS-2}, \quad i \in \{ i : \lambda_i > \lambda \} \\
\sum_{i=1}^{n} x_i^2 & \leq 1 \\
\sum_{i \in \{i: \lambda_i > \lambda \}} \xi_i + \sum_{i \in \{i: \lambda_i \leq \lambda \}} g_i^2 & \leq 1 + \frac{1}{4N^2} \sum_{i \in \{i: \lambda_i > \lambda \}} \theta_i^2 \\
\sum_{i=1}^{n} y_i & \leq \sqrt{k} & \quad (\|x\|_1 \leq \sqrt{k}) \\
-\theta_i & \leq g_i \leq \theta_i, & \quad i \in [n] \\
-\lambda_i & \leq \lambda_i \leq \lambda, & \quad i \in \{ i : \lambda_i \leq \lambda \} \\
(\text{Convex-IP})
\end{align*}
\]

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 \( (\gamma_i^j)_{j=-N}^{N} \leftarrow \left( \frac{k}{N} \cdot \theta_i \right)_{j=-N}^{N} \) be the value of the \( j^{th} \) splitting point.

Since \( v_i \)'s are orthogonal, \( \sum_{i=1}^{n} x_i^2 \leq 1 \) implies \( \sum_{i=1}^{n} g_i^2 \leq 1 \). Then together with \( \xi_i \) representing \( g_i^2 \), we can obtain the implied inequality \( \sum_{i \in \{i: \lambda_i > \lambda \}} \xi_i + \sum_{i \in \{i: \lambda_i \leq \lambda \}} g_i^2 \leq 1 + \frac{1}{4N^2} \sum_{i \in \{i: \lambda_i > \lambda \}} \theta_i^2 \).

The extra term in the right-hand-side reflects the fact that \( \xi_i \) is not exactly equal to \( g_i^2 \), but only a piecewise linear upper bound on \( g_i^2 \). In fact, we have that \( \xi_i \leq g_i^2 + \frac{\theta_i^2}{4N^2} \). 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 $OPT_{\text{convex-IP}}$ 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 $\ell_1$-relax by Convex-IP, we obtain the following result:

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

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

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 $\lambda$, 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 $OPT_{\text{convex-IP}}$ 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 $\lambda$ 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 $\lambda \triangleq \max\{\lambda_i: \lambda_i \leq \lambda\} = \lambda$, one can apply Algorithm 1 to obtain a matrix $\tilde{A} \succeq A$, such that $\tilde{\lambda} < \lambda$. We then replace $A$ by $\tilde{A}$.

Algorithm 1 Perturbation of $A$

1: Input: Sample covariance matrix $A$ and $\lambda$.
2: Output: A perturbed sample covariance matrix $\tilde{A}$ with distinct eigenvalues such that $\tilde{A} \succeq A$.
3: function PERTURBATION$(A, \lambda)$
4: Compute spectral decomposition on $A$ as $A = V^T \Lambda V$ and let $\lambda_1 > \cdots > \lambda = \lambda_j > \cdot \lambda_p \geq 0$
5: be all its distinct values of eigenvalues where $p \leq n$.
6: Set $\Delta \lambda \leftarrow \min\{\lambda_i - \lambda_{i+1}: i = 1, \ldots, p - 1\}$.
7: Set $\Delta \lambda \leftarrow \lambda + \text{diag}(\frac{\Delta \lambda}{n} \epsilon: i = n, \ldots, 1)$ with $\epsilon < \Delta \lambda$.
8: return $\tilde{A} \leftarrow V^T \tilde{\Lambda} V$.
2. Perturb the covariance matrix \( A = \sum_{i=1}^n \lambda_i v_i v_i^\top \) by \( \tilde{A} = \sum_{i \in \{i: \lambda_i > \lambda\}} \lambda_i v_i v_i^\top + \sum_{i \in \{i: \lambda_i \leq \lambda\}} \tilde{\lambda} v_i v_i^\top \). Note that the objective value \( \text{OPT}_{\text{convex-IP}}(A) \) in Convex-IP is an upper bound on \( \text{OPT}_{\text{convex-IP}}(A) \). Replace \( A \) by \( \tilde{A} \).

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

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

\[
1 \leq \sum_{i \in \{i: \lambda_i > \lambda\}} \xi_i + \sum_{i \in \{i: \lambda_i \leq \lambda\}} \tilde{g}_i^2 \leq 1 + \frac{1}{4N^2} \sum_{i \in \{i: \lambda_i > \lambda\}} \theta_i^2,
\]

\[
\sum_{i=1}^n \tilde{g}_i^2 = \sum_{i \in \{i: \lambda_i > \lambda\}} \tilde{g}_i^2 + \sum_{i \in \{i: \lambda_i \leq \lambda\}} \tilde{g}_i^2 \leq 1,
\]

imply the following inequalities:

\[
1 - \frac{s}{\lambda - \lambda} \leq \sum_{i \in \{i: \lambda_i > \lambda\}} \xi_i \leq 1 + \frac{1}{4N^2} \sum_{i \in \{i: \lambda_i > \lambda\}} \theta_i^2 - \frac{s}{\lambda - \lambda},
\]

\[
\sum_{i \in \{i: \lambda_i > \lambda\}} \tilde{g}_i^2 \leq 1 - \frac{s}{\lambda - \lambda}.
\]

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

\[
\begin{align*}
\lambda + \max_{x,y,g,\xi,\eta,s} & \left( \lambda_i - \lambda \right) \xi_i - s \triangleq \text{OPT}_{\text{Pert-Convex-IP}} \\
g_i &= x^\top v_i, \quad i \in \{i: \lambda_i > \lambda\} \\
g_i &= \sum_{j=-N}^N \gamma_i^j \eta_i^j, \quad i \in \{i: \lambda_i > \lambda\} \\
\xi_i &= \sum_{j=-N}^N (\gamma_i^j)^2 \eta_i^j, \quad i \in \{i: \lambda_i > \lambda\} \\
(\eta_i^{-N}, \ldots, \eta_i^N) & \in \text{SOS-2}, \quad i \in \{i: \lambda_i > \lambda\}
\end{align*}
\]

s.t.

\[
\begin{align*}
\sum_{i=1}^n x_i^2 & \leq 1 \\
\sum_{i \in \{i: \lambda_i > \lambda\}} g_i^2 & \leq 1 - \frac{s}{\lambda - \lambda} \\
1 - \frac{s}{\lambda - \lambda} & \leq \sum_{i \in \{i: \lambda_i > \lambda\}} \xi_i \leq 1 + \frac{1}{4N^2} \sum_{i \in \{i: \lambda_i > \lambda\}} \theta_i^2 - \frac{s}{\lambda - \lambda}
\end{align*}
\]

\[
\sum_{i=1}^n y_i \leq \sqrt{k} \quad \text{and} \quad y_i \geq |x_i|, \quad i \in \{i: \lambda_i > \lambda\} \quad \text{(||x||}_2 \leq 1)
\]

\[
-\theta_i \leq g_i \leq \theta_i, \quad i \in \{i: \lambda_i > \lambda\} \quad \text{(||x||}_1 \leq \sqrt{k})
\]

\[
g_i \leq \theta_i, \quad i \in [n] \quad \text{(||g||}_1 \leq \theta_i, \quad i \in [n])
\]

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

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

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 $\xi_i$ 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:

$$v_{ij} = \begin{cases} |x_{ij}| & \text{if } j \leq k \\ |x_{ik}| & \text{if } j > k. \end{cases} \quad (8)$$

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

$$v^\top y \leq b_v, \quad (9)$$

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 $\ell_1$-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

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

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

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:

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

Without loss of generality, we assume that \( x \geq 0 \) and \( [x]_1 \geq [x]_2 \geq \ldots \geq [x]_n \geq 0 \). 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 \( \|\bar{v}^1\|_2 \geq \ldots \geq \|\bar{v}^m\|_2 \). Let \( I^1, \ldots, I^m \) 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

\[ [x_{I^1}]_j \geq [x_{I^2}]_2 \geq \ldots \geq [x_{I^g}]_g \]

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

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

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

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

Now repeating the above step, we obtain the result. \( \square \)

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 \( \rho \) be obtained by solving the following optimization problem:

\[
\rho = \max_x \|x_{I^1}\|_2 + \cdots + \|x_{I^m}\|_2 \\
\text{s.t.} \quad x = x_{I^1} + \cdots + x_{I^m} \\
\|x\|_2 = \|x_{I^1}\|_2 + \cdots + \|x_{I^m}\|_2 \leq 1 \\
\|x\|_1 = \|x_{I^1}\|_1 + \cdots + \|x_{I^m}\|_1 \leq \sqrt{k} \\
[x]_1 \geq \cdots \geq [x]_n \geq 0. \quad \text{(Approximation ratio)}
\]


Then we obtain

\[ T_k \subseteq \rho \cdot \text{Conv} (S_k). \] (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 \( \rho \) 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:

\[
\theta = \max \ u + v \quad \text{subject to} \quad u^2 + v^2 \leq 1
\]

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 \( \rho \). 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 \( \ell_1 \) and \( \ell_2 \) 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

\[ \|x_{I_2}\|_2 \leq \sqrt{\left( \|x_{I_1}\|_1/k \right)^2 k} = \frac{t}{\sqrt{k}}. \] (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

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

Therefore we obtain

\[
\sum_{i=1}^{m} \|x_{I_i}\|_2 = \|x_{I_1}\|_2 + \|x_{I_2}\|_2 + \sum_{i=3}^{m} \|x_{I_i}\|_2 \leq \gamma + \min \left\{ \frac{t}{\sqrt{k}}, \sqrt{1 - \gamma^2} \right\} + 1 - \frac{t}{\sqrt{k}}. \quad \text{(Upper-Bound)}
\]

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 \( \gamma \geq \frac{1}{\sqrt{2}} \geq \sqrt{1 - \gamma^2} \), \( \gamma \) 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
γ ≤ \sqrt{\frac{k}{k+1}} we obtain two cases:

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

\[
\leq \begin{cases}
1 + \sqrt{\frac{k}{k+1}} & \\
1 + \frac{k}{k+1}
\end{cases}
\]

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 \( \gamma \) 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.

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

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 \( I^0 \leftarrow I_k(V^T x^0) \) where \( V \) is a square root of \( A \), i.e. \( A = V^T V \). In the \( i \)-th iteration, we update

\[
I^i \leftarrow I_k(V^T x^i), \quad x^{i+1} \leftarrow \arg \max_{\|x\|_2 = 1} x^\top A_{I^i} x
\]

(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 \( \|x\|_0 \leq k \). Moreover, using the fact \( A \) is a PSD matrix, it is easy to verify that \((x_i + 1)^\top A x_i + 1 \geq (x_i)^\top A x_i \) 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**

1. **Input:** Sample covariance matrix \( A \), cardinality constraint \( k \), initial vector \( x^0 \).
2. **Output:** A feasible solution \( x^* \) of SPCA, and its objective value.
3. **function** HEURISTIC METHOD\( (A, k, x^0) \)
   4. Start with an initial (randomized) vector \( x^0 \) such that \( \|x^0\|_2 = 1 \) and \( \|x^0\|_0 \leq k \).
   5. Set the initial current objective value \( \text{Obj} \leftarrow (x^0)^\top A x^0 \).
   6. Set the initial past objective value \( \tilde{\text{Obj}} \leftarrow 0 \).
   7. Set the maximum number of iterations be \( i^{\text{max}} \).
   8. **while** \( \text{Obj} - \tilde{\text{Obj}} > \epsilon \) and \( i \leq i^{\text{max}} \) **do**
      9. Set \( \text{Obj} \leftarrow \tilde{\text{Obj}} \).
      10. Set \( I_i \leftarrow I_k( V^\top x^i ) \).
      11. Set \( x^{i+1} \leftarrow \arg \max_{x \in \mathbb{R}^n} \|x\|_0 = 1 x^\top A I_i x \).
      12. Set \( \text{Obj} \leftarrow (x^{i+1})^\top A x^{i+1} \).
   **end while**
13. **return** \( x^* \) as the final \( x \) obtained from while-loop, and \( \text{Obj} \).
15. **end function**

We use the values of \( \epsilon = 10^{-6} \) and \( i^{\text{max}} = 20 \) 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 \( \max \{ x^\top A y \| x\|_2 = \|y\|_2 = 1, \|x\|_0 \leq k, \|y\|_0 \leq k \} \) — 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 \( \lambda, \quad N \)

1. **Deciding \( \lambda \):** The size of the set \( \{ i : \lambda_i > \lambda \} \) denoted by \( I_{\text{pos}} \) plays an important role for the computational tractability of our method. So our algorithm inputs an initial value, \( I_{\text{pos}}^{\text{ini}} \). From the primal heuristic, we obtain a lower bound \( \text{LB}_{\text{primal}}^{\lambda_k} \) on \( \lambda^k(A) \). Let
   \[
   \lambda_{i_1} \geq \lambda_{i_2} \geq \cdots \geq \lambda_{i_n},
   \]


be the eigenvalues of $A$. If $\lambda_{ini} < LB_{primal}$, then we set $\lambda \triangleq \lambda_{ini}$. On the other hand, if $\lambda_{ini} > LB_{primal}$, then let $l$ be the smallest index such that $\lambda_l > LB_{primal}$ and we set $\lambda \triangleq \lambda_l$.

2. Deciding $N$: In practice, $\theta_i$ 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: $(r_{ini}, N) = (10, 3)$. For the Pert-Convex-IP method, we let “iter” denote the maximum number of iterations. We used three settings in our experiments:

$$(r_{ini}, N, \text{iter}) \in \{(5, 3, 10), (10, 3, 3), (15, 3, 2)\}.$$  

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 $\ell_1$ 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 $\ell_1$-relax problem. We compute a good feasible solution of $\ell_1$-relax problem $\max ||x||_2 \leq 1, ||x||_1 \leq \sqrt{k} x^T A x$ via a heuristic method similar to Algorithm 2. Details are presented in Appendix F. We also find a dual bound to $\ell_1$-relax (note that the cutting-planes we added to Pert-Convex-IP, and the inequalities $- \theta_i \leq g_i \leq \theta_i$ are not valid for $\ell_1$ relaxation) by solving an appropriate variant of Pert-Convex-IP ($\theta_i$’s are computed so that they are valid for the $\ell_1$ 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 (size, cardinality) denoting the number of columns/rows of the A matrix and the right-hand-side of the $\ell_0$ constraint of the original SPCA problem.

- **LB-$\ell_0$**: denotes the lower bound on the SPCA problem obtained from the (heuristic) Algorithm 2 in Section 4.2.

- **LB-$\ell_1$**: denotes the lower bound on the $\ell_1$-relax problem obtained from the heuristic method presented in Appendix F.

- **Convex-IP-$\ell_0$, Pert-Convex-IP-$\ell_0$, Pert-Convex-IP-$\ell_1$**: denote the convex integer program which is a relaxation of SPCA problem, the Pert-Convex-IP-$\ell_0$ is a relaxation of SPCA problem, the Pert-Convex-IP-$\ell_1$ is a relaxation of $\ell_1$-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-$\ell_0$, Pert-Convex-IP-$\ell_0$, Pert-Convex-IP-$\ell_1$, SDP).

- **gap**: denotes the approximation ratio (duality gap) obtained by the formula $\text{gap} = \frac{\text{UB} - \text{LB-$\ell_0$}}{\text{LB-$\ell_0$}}$ or $\text{gap} = \frac{\text{UB} - \text{LB-$\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: $(I_{pos}, N, \text{iter}) = \{(5, 3, 10), (10, 3, 3), (15, 3, 2)\}$.

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 $300 \times 300$.
   
   - **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 $500 \times 500$.
   
   - **Pert-Convex-IP**: The perturbed convex IP scales significantly better that the other methods. While we experimented with instances up to size $2000 \times 2000$, 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-ℓ₀ should be smaller than Pert-Convex-IP-ℓ₁, if we solve to optimality. But in some cases, Pert-Convex-IP-ℓ₁ may run faster than Pert-Convex-IP-ℓ₀, 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 than 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-ℓ₀ can be explained on the basis of \( \ell_1 \)-relaxation. (Compare for example the upper bound from Pert-Convex-IP-ℓ₀ to the lower bound on \( \ell_1 \) 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>
<thead>
<tr>
<th>Case</th>
<th>LB-$f_0$</th>
<th>Convex-IP-$f_0$</th>
<th>Pert-Convex-IP-$f_0$</th>
<th>SDP</th>
<th>LB-$f_1$</th>
<th>Pert-Convex-IP-$f_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LR</td>
<td>gap</td>
<td>LR</td>
<td>gap</td>
<td>LR</td>
<td>gap</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>511.95</td>
<td>511.98</td>
<td>0.005%</td>
<td>380</td>
<td>511.95</td>
<td>511.98</td>
</tr>
<tr>
<td>Case 2</td>
<td>522.45</td>
<td>522.47</td>
<td>0.003%</td>
<td>469</td>
<td>522.45</td>
<td>522.47</td>
</tr>
<tr>
<td>Case 1</td>
<td>414.04</td>
<td>414.15</td>
<td>0.027%</td>
<td>1692</td>
<td>414.11</td>
<td>414.16</td>
</tr>
<tr>
<td></td>
<td>568.56</td>
<td>568.62</td>
<td>0.011%</td>
<td>1067</td>
<td>568.65</td>
<td>568.64</td>
</tr>
<tr>
<td></td>
<td>478.24</td>
<td>478.36</td>
<td>0.025%</td>
<td>2598</td>
<td>478.44</td>
<td>478.39</td>
</tr>
<tr>
<td></td>
<td>420.91</td>
<td>421.03</td>
<td>0.037%</td>
<td>3374</td>
<td>421.13</td>
<td>421.12</td>
</tr>
<tr>
<td></td>
<td>256.82</td>
<td>257.24</td>
<td>0.164%</td>
<td>7525</td>
<td>257.34</td>
<td>257.29</td>
</tr>
<tr>
<td></td>
<td>551.74</td>
<td>551.90</td>
<td>0.029%</td>
<td>7186</td>
<td>551.97</td>
<td>551.95</td>
</tr>
<tr>
<td>Case 1</td>
<td>415.16</td>
<td>NaN</td>
<td>-</td>
<td>1147</td>
<td>317.09</td>
<td>316.86</td>
</tr>
<tr>
<td></td>
<td>383.14</td>
<td>NaN</td>
<td>-</td>
<td>3643</td>
<td>384.74</td>
<td>384.66</td>
</tr>
</tbody>
</table>

Table 1: Spiked Covariance Recovery - Cardinality 10
<table>
<thead>
<tr>
<th>Case</th>
<th>LB-(\ell_0) Convex-IP-(\ell_0) Pert-Convex-IP-(\ell_0) SDP</th>
<th>LB-(\ell_1) Pert-Convex-IP-(\ell_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UB</td>
<td>gap</td>
</tr>
<tr>
<td>Case 1 (200, 20)</td>
<td>516</td>
<td>526.56</td>
</tr>
<tr>
<td>Case 2 (200, 20)</td>
<td>593</td>
<td>598.84</td>
</tr>
<tr>
<td>Case 1 (300, 20)</td>
<td>499</td>
<td>502.48</td>
</tr>
<tr>
<td>Case 2 (300, 20)</td>
<td>600</td>
<td>606.76</td>
</tr>
<tr>
<td>Case 1 (400, 20)</td>
<td>483</td>
<td>496.24</td>
</tr>
<tr>
<td>Case 2 (400, 20)</td>
<td>428</td>
<td>436.20</td>
</tr>
<tr>
<td>Case 1 (500, 20)</td>
<td>294</td>
<td>297.51</td>
</tr>
<tr>
<td>Case 2 (500, 20)</td>
<td>591</td>
<td>582.18</td>
</tr>
<tr>
<td>Case 1 (1000, 20)</td>
<td>414</td>
<td>-</td>
</tr>
<tr>
<td>Case 2 (1000, 20)</td>
<td>391</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3: Synthetic Example - Cardinality 10

<table>
<thead>
<tr>
<th>Case</th>
<th>LB-(\ell_0)</th>
<th>Convex-IP-(\ell_0)</th>
<th>Pert-Convex-IP-(\ell_0)</th>
<th>SDP</th>
<th>LB-(\ell_1)</th>
<th>Pert-Convex-IP-(\ell_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UB gap</td>
<td>Time</td>
<td>UB gap</td>
<td>Time</td>
<td>UB gap</td>
<td>Time</td>
</tr>
<tr>
<td>Case 1 (200, 10)</td>
<td>5634.14</td>
<td>6227.54</td>
<td>10.5 %</td>
<td>6000</td>
<td>5642.24</td>
<td>0.14 %</td>
</tr>
<tr>
<td></td>
<td>6064</td>
<td>0.14 %</td>
<td>38</td>
<td>5643</td>
<td>0.07 %</td>
<td>44</td>
</tr>
<tr>
<td>Case 2 (200, 10)</td>
<td>7321.22</td>
<td>7961.8</td>
<td>8.75 %</td>
<td>6000</td>
<td>7330.92</td>
<td>0.13 %</td>
</tr>
<tr>
<td></td>
<td>7333</td>
<td>0.05 %</td>
<td>23</td>
<td>7332</td>
<td>0.07 %</td>
<td>35</td>
</tr>
<tr>
<td>Case 1 (300, 10)</td>
<td>4157.46</td>
<td>4423.74</td>
<td>6.40 %</td>
<td>6000</td>
<td>4168.86</td>
<td>0.27 %</td>
</tr>
<tr>
<td></td>
<td>4162</td>
<td>0.19 %</td>
<td>21</td>
<td>4169</td>
<td>0.16 %</td>
<td>47</td>
</tr>
<tr>
<td>Case 2 (300, 10)</td>
<td>5135.48</td>
<td>5144.96</td>
<td>0.18 %</td>
<td>6000</td>
<td>5147.40</td>
<td>0.23 %</td>
</tr>
<tr>
<td></td>
<td>5147</td>
<td>0.14 %</td>
<td>13</td>
<td>5148</td>
<td>0.12 %</td>
<td>111</td>
</tr>
<tr>
<td>Case 1 (400, 10)</td>
<td>6519.36</td>
<td>6608.10</td>
<td>1.36 %</td>
<td>4762</td>
<td>6533.74</td>
<td>0.22 %</td>
</tr>
<tr>
<td></td>
<td>6534</td>
<td>0.18 %</td>
<td>44</td>
<td>6536</td>
<td>0.15 %</td>
<td>74</td>
</tr>
<tr>
<td>Case 2 (400, 10)</td>
<td>5942.04</td>
<td>6003.90</td>
<td>1.04 %</td>
<td>6000</td>
<td>5963.72</td>
<td>0.36 %</td>
</tr>
<tr>
<td></td>
<td>5965</td>
<td>0.17 %</td>
<td>29</td>
<td>5973</td>
<td>0.13 %</td>
<td>331</td>
</tr>
<tr>
<td>Case 1 (500, 10)</td>
<td>5125.84</td>
<td>5227.90</td>
<td>1.98 %</td>
<td>6000</td>
<td>5145.96</td>
<td>0.38 %</td>
</tr>
<tr>
<td></td>
<td>5144</td>
<td>0.29 %</td>
<td>44</td>
<td>5146</td>
<td>0.25 %</td>
<td>169</td>
</tr>
<tr>
<td>Case 2 (500, 10)</td>
<td>5534.84</td>
<td>5517.78</td>
<td>1.30 %</td>
<td>6000</td>
<td>5566.36</td>
<td>0.39 %</td>
</tr>
<tr>
<td></td>
<td>5560</td>
<td>0.23 %</td>
<td>40</td>
<td>5571</td>
<td>0.20 %</td>
<td>827</td>
</tr>
<tr>
<td>Case 1 (1000, 10)</td>
<td>NaN</td>
<td>-</td>
<td>-</td>
<td></td>
<td>5145</td>
<td>0.58 %</td>
</tr>
<tr>
<td></td>
<td>5149</td>
<td>0.49 %</td>
<td>128</td>
<td>5146</td>
<td>0.43 %</td>
<td>3043</td>
</tr>
<tr>
<td>Case 2 (1000, 10)</td>
<td>NaN</td>
<td>-</td>
<td>-</td>
<td></td>
<td>6973.33</td>
<td>0.39 %</td>
</tr>
<tr>
<td></td>
<td>6974</td>
<td>0.39 %</td>
<td>129</td>
<td>6969</td>
<td>0.32 %</td>
<td>1538</td>
</tr>
</tbody>
</table>
Table 4: Synthetic Example - Cardinality 20

<table>
<thead>
<tr>
<th>Case</th>
<th>LB-ℓ₀</th>
<th>UB</th>
<th>gap</th>
<th>Time</th>
<th>UB</th>
<th>gap</th>
<th>Time</th>
<th>UB</th>
<th>gap</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1 (200, 20)</td>
<td>11221</td>
<td>11278</td>
<td>0.50%</td>
<td>271</td>
<td>11227</td>
<td>0.04%</td>
<td>66</td>
<td>11226</td>
<td>0.04%</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (200, 20)</td>
<td>14588</td>
<td>14643</td>
<td>0.38%</td>
<td>236</td>
<td>14593</td>
<td>0.03%</td>
<td>90</td>
<td>14593</td>
<td>0.03%</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1 (300, 20)</td>
<td>8282</td>
<td>8406</td>
<td>1.50%</td>
<td>663</td>
<td>8290</td>
<td>0.10%</td>
<td>128</td>
<td>8292</td>
<td>0.12%</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (300, 20)</td>
<td>10233</td>
<td>10264</td>
<td>0.30%</td>
<td>676</td>
<td>10242</td>
<td>0.09%</td>
<td>127</td>
<td>10240</td>
<td>0.07%</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1 (400, 20)</td>
<td>12976</td>
<td>13166</td>
<td>1.46%</td>
<td>1680</td>
<td>12987</td>
<td>0.08%</td>
<td>166</td>
<td>12986</td>
<td>0.08%</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (400, 20)</td>
<td>11809</td>
<td>11945</td>
<td>1.15%</td>
<td>2636</td>
<td>11819</td>
<td>0.08%</td>
<td>199</td>
<td>11818</td>
<td>0.08%</td>
<td>485</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1 (500, 20)</td>
<td>10218</td>
<td>10419</td>
<td>1.97%</td>
<td>3349</td>
<td>10231</td>
<td>0.13%</td>
<td>314</td>
<td>10231</td>
<td>0.13%</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (500, 20)</td>
<td>11032</td>
<td>11188</td>
<td>1.41%</td>
<td>3646</td>
<td>11045</td>
<td>0.12%</td>
<td>265</td>
<td>11044</td>
<td>0.11%</td>
<td>124</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1 (1000, 20)</td>
<td>10191</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>10191</td>
<td>0.26%</td>
<td>745</td>
<td>10197</td>
<td>0.24%</td>
<td>483</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (1000, 20)</td>
<td>13867</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>13894</td>
<td>0.19%</td>
<td>932</td>
<td>13892</td>
<td>0.18%</td>
<td>8807</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 5: Controlling Sparsity - Cardinality 10

<table>
<thead>
<tr>
<th>Case</th>
<th>LB-f₀</th>
<th>Convex-IP-f₀</th>
<th>Pert-Convex-IP-f₀</th>
<th>SDP-f₀</th>
<th>LB-f₁</th>
<th>Pert-Convex-IP-f₁</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UB</td>
<td>gap</td>
<td>Time</td>
<td>UB</td>
<td>gap</td>
<td>Time</td>
</tr>
<tr>
<td>Case 1 (200, 10)</td>
<td>706</td>
<td>0.14 %</td>
<td>925</td>
<td>727</td>
<td>2.9 %</td>
<td>117</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (200, 10)</td>
<td>680</td>
<td>0.14 %</td>
<td>1195</td>
<td>704</td>
<td>5.5 %</td>
<td>176</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1 (300, 10)</td>
<td>972</td>
<td>1.4 %</td>
<td>1958</td>
<td>1010</td>
<td>3.91 %</td>
<td>135</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (300, 10)</td>
<td>976</td>
<td>1.1 %</td>
<td>3007</td>
<td>1013</td>
<td>3.79 %</td>
<td>278</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1 (400, 10)</td>
<td>1239</td>
<td>1.3 %</td>
<td>7207</td>
<td>1283</td>
<td>2.11 %</td>
<td>709</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (400, 10)</td>
<td>1207</td>
<td>1.6 %</td>
<td>7220</td>
<td>1250</td>
<td>3.56 %</td>
<td>221</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1 (500, 10)</td>
<td>1498</td>
<td>2.1 %</td>
<td>12180</td>
<td>1576</td>
<td>4.21 %</td>
<td>1026</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2 (500, 10)</td>
<td>1498</td>
<td>2.1 %</td>
<td>1397</td>
<td>1560</td>
<td>4.14 %</td>
<td>251</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1 (1000, 10)</td>
<td>3948</td>
<td>NaN</td>
<td>-</td>
<td>6305</td>
<td>2.97 %</td>
<td>2036</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6052</td>
<td>53.3 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5902</td>
<td>49.5 %</td>
<td></td>
</tr>
<tr>
<td>Case 2 (1000, 10)</td>
<td>4002</td>
<td>NaN</td>
<td>-</td>
<td>6325</td>
<td>51.0 %</td>
<td>3270</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6049</td>
<td>47.6 %</td>
<td></td>
</tr>
</tbody>
</table>

### Table 6: Controlling Sparsity - Cardinality 20

<table>
<thead>
<tr>
<th>Case</th>
<th>LB-f₀</th>
<th>Convex-IP-f₀</th>
<th>Pert-Convex-IP-f₀</th>
<th>SDP-f₀</th>
<th>LB-f₁</th>
<th>Pert-Convex-IP-f₁</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UB</td>
<td>gap</td>
<td>Time</td>
<td>UB</td>
<td>gap</td>
<td>Time</td>
</tr>
<tr>
<td>Case 1 (200, 20)</td>
<td>1341</td>
<td>0.97 %</td>
<td>277</td>
<td>1341</td>
<td>0.04 %</td>
<td>1288</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1341.6</td>
<td>0.04 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1342.0</td>
<td>0.07 %</td>
<td></td>
</tr>
<tr>
<td>Case 2 (200, 20)</td>
<td>1287</td>
<td>1.63 %</td>
<td>332</td>
<td>1287</td>
<td>0.04 %</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1287.6</td>
<td>0.04 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1287.6</td>
<td>0.04 %</td>
<td></td>
</tr>
<tr>
<td>Case 1 (300, 20)</td>
<td>1839</td>
<td>1.25 %</td>
<td>1019</td>
<td>1830</td>
<td>0.44 %</td>
<td>252</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1848.8</td>
<td>0.53 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1850.6</td>
<td>0.63 %</td>
<td></td>
</tr>
<tr>
<td>Case 2 (300, 20)</td>
<td>1849</td>
<td>1.35 %</td>
<td>707</td>
<td>1852</td>
<td>0.19 %</td>
<td>362</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1854.0</td>
<td>0.27 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1854.0</td>
<td>0.27 %</td>
<td></td>
</tr>
<tr>
<td>Case 1 (400, 20)</td>
<td>2339</td>
<td>1.45 %</td>
<td>907</td>
<td>2388.7</td>
<td>2.12 %</td>
<td>287</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2387.9</td>
<td>2.04 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2387.9</td>
<td>2.09 %</td>
<td></td>
</tr>
<tr>
<td>Case 2 (400, 20)</td>
<td>2301</td>
<td>2327 %</td>
<td>3106</td>
<td>2350</td>
<td>2.35 %</td>
<td>452</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2348.9</td>
<td>2.08 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2348.9</td>
<td>2.08 %</td>
<td></td>
</tr>
<tr>
<td>Case 1 (500, 20)</td>
<td>2658</td>
<td>2.34 %</td>
<td>2773</td>
<td>2966.8</td>
<td>3.81 %</td>
<td>725</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2961.6</td>
<td>3.62 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2962.5</td>
<td>3.66 %</td>
<td></td>
</tr>
<tr>
<td>Case 2 (500, 20)</td>
<td>2832</td>
<td>2.37 %</td>
<td>3015</td>
<td>2831.9</td>
<td>3.53 %</td>
<td>455</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2928.7</td>
<td>3.41 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2928.8</td>
<td>3.42 %</td>
<td></td>
</tr>
<tr>
<td>Case 1 (1000, 20)</td>
<td>7535</td>
<td>- %</td>
<td>9277</td>
<td>9870</td>
<td>31.0 %</td>
<td>1907</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9642</td>
<td>28.0 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Case 2 (1000, 20)</td>
<td>7789</td>
<td>- %</td>
<td>9277</td>
<td>10087</td>
<td>22.4 %</td>
<td>4009</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9717</td>
<td>21.4 %</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9717</td>
<td>21.4 %</td>
<td></td>
</tr>
</tbody>
</table>
Table 7: First six sparse principal components of Pitprops

<table>
<thead>
<tr>
<th>Cardinality</th>
<th>LB-$\ell_0$</th>
<th>Convex-IP-$\ell_0$</th>
<th>Pert-Convex-IP-$\ell_0$</th>
<th>SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UB</td>
<td>gap</td>
<td>Time</td>
<td>UB</td>
</tr>
<tr>
<td>Cardinality 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.517</td>
<td>3.2 %</td>
<td>0.40</td>
<td>3.611</td>
<td>6.0 %</td>
</tr>
<tr>
<td>Cardinality 2</td>
<td>1.882</td>
<td>1.909</td>
<td>1.4 %</td>
<td>1.949</td>
</tr>
<tr>
<td>Cardinality 1</td>
<td>1.364</td>
<td>1.417</td>
<td>3.8 %</td>
<td>1.408</td>
</tr>
<tr>
<td>Cardinality 1</td>
<td>1.012</td>
<td>1.022</td>
<td>1.2 %</td>
<td>1.035</td>
</tr>
<tr>
<td>Sum of above</td>
<td>9.652</td>
<td>9.897</td>
<td>2.5 %</td>
<td>10.12</td>
</tr>
</tbody>
</table>

Table 8: Biological and Internet Data - Cardinality 10

<table>
<thead>
<tr>
<th>Case</th>
<th>LB-$\ell_0$</th>
<th>Convex-IP-$\ell_0$</th>
<th>Pert-Convex-IP-$\ell_0$</th>
<th>SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UB</td>
<td>gap</td>
<td>Time</td>
<td>UB</td>
</tr>
<tr>
<td>Eisen-1 (79, 10)</td>
<td>17.33</td>
<td>17.39</td>
<td>0.3 %</td>
<td>17.35</td>
</tr>
<tr>
<td>Eisen-2 (118, 10)</td>
<td>11.71</td>
<td>11.87</td>
<td>1.4 %</td>
<td>12.19</td>
</tr>
<tr>
<td>Colon (500, 10)</td>
<td>2641</td>
<td>3028</td>
<td>14.7 %</td>
<td>3333</td>
</tr>
<tr>
<td>Lymphoma (500, 10)</td>
<td>6008</td>
<td>7583</td>
<td>20.7 %</td>
<td>8470</td>
</tr>
<tr>
<td>Reddit (2000, 10)</td>
<td>1523</td>
<td>NaN</td>
<td>-</td>
<td>1712</td>
</tr>
</tbody>
</table>

5 Acknowledgements

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

References


Table 9: Biological and Internet Data - Cardinality 20

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Eisen-1 (79, 20)</td>
<td>17.71</td>
<td>17.94</td>
<td>1.40 %</td>
<td>742</td>
<td>17.73</td>
<td>0.11 %</td>
<td>1.30 %</td>
<td>757</td>
<td>0.11 %</td>
<td>757</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17.74</td>
<td>0.17 %</td>
<td>1.30 %</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eisen-2 (118, 20)</td>
<td>19.32</td>
<td>19.71</td>
<td>2.02 %</td>
<td>64</td>
<td>19.57</td>
<td>1.29 %</td>
<td>200</td>
<td>149</td>
<td>1.35 %</td>
<td>757</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.46</td>
<td>0.72 %</td>
<td>19.76</td>
<td>54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Colon (500, 20)</td>
<td>4255</td>
<td>4907</td>
<td>15.3 %</td>
<td>7280</td>
<td>4950</td>
<td>16.5 %</td>
<td>1625</td>
<td>NaN</td>
<td>- %</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4500</td>
<td>5.76 %</td>
<td>7285</td>
<td>2795</td>
<td>- %</td>
<td>-</td>
</tr>
<tr>
<td>Lymphoma (500, 20)</td>
<td>9882</td>
<td>10983</td>
<td>18.7 %</td>
<td>7289</td>
<td>11123</td>
<td>22.8 %</td>
<td>971</td>
<td>NaN</td>
<td>- %</td>
<td>-</td>
</tr>
<tr>
<td>Reddit (2000, 20)</td>
<td>1119</td>
<td>-</td>
<td>- %</td>
<td>-</td>
<td>1151</td>
<td>2.86 %</td>
<td>4735</td>
<td>NaN</td>
<td>- %</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1150</td>
<td>2.77 %</td>
<td>8628</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


A SDP relaxation

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

\[
\begin{align*}
\max & \quad \text{tr}(AX) \\
\text{s.t.} & \quad \text{tr}(X) = 1, \|X\|_0 \leq k^2, X \succeq 0, \text{rank}(X) = 1.
\end{align*}
\]

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

\[
\begin{align*}
\max & \quad \text{tr}(AX) \\
\text{s.t.} & \quad \text{tr}(X) = 1, 1^\top X 1 \leq k, X \succeq 0. \quad \text{(SDP)}
\end{align*}
\]

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

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

24
Note that the above solution \((\bar{x}, \bar{y}, \bar{g}, \bar{\xi}, \bar{\eta}, \bar{s})\) is a feasible solution for Convex-IP. This is easy to verify for all the constraints except the constraint \(\sum_{i \in \{\lambda_i > \lambda\}} \xi_i + \sum_{i \in \{\lambda_i \leq \lambda\}} g_i^2 \leq 1 + \frac{1}{4N^2} \sum_{i \in \{\lambda_i > \lambda\}} \theta_i^2\). 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 \((\bar{x}, \bar{y}, \bar{g}, \bar{\xi}, \bar{\eta}, \bar{s})\) is

\[
\lambda + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)\xi_i^* - s^* \geq \lambda + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)(g_i^{*})^2 - s^* = \lambda + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)((x^*)^Tv_i)^2 + \sum_{i \in \{\lambda_i \leq \lambda\}} (\lambda_i - \lambda)((x^*)^Tv_i)^2 = \lambda + \sum_{i=1}^{n} (\lambda_i - \lambda)((x^*)^Tv_i)^2.
\]

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

\[\square\]

C Proof of Proposition 2

\textit{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:

\[
\text{OPT}_{\text{convex-IP}} = \lambda + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)\bar{\xi}_i - \bar{s} = \lambda + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)(\bar{\xi}_i - \bar{g}_i^2 + \bar{g}_i^2) - \bar{s} = \lambda + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)(\bar{\xi}_i - \bar{g}_i^2) + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)\bar{g}_i^2 - \bar{s}.
\]

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

\[
\lambda + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)(\bar{\xi}_i - \bar{g}_i^2) = \lambda + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)(\bar{\xi}_i - \bar{g}_i^2) + \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)\bar{g}_i^2 + \sum_{i \in \{\lambda_i \leq \lambda\}} (\lambda_i - \lambda)\bar{g}_i^2 \]

\[
= \sum_{i \in \{\lambda_i > \lambda\}} (\lambda_i - \lambda)(\bar{\xi}_i - \bar{g}_i^2) + \left(\lambda + \sum_{i=1}^{n} (\lambda_i - \lambda)\bar{g}_i^2\right).
\]

By previous results, \(\lambda + \sum_{i=1}^{n} (\lambda_i - \lambda)\bar{g}_i^2 = \bar{x}^T\bar{A}\bar{x}\). Note that due to the \(\ell_2\)-norm constraint \(\|x\|_2 \leq 1\) and the \(\ell_1\)-norm constraint present in Convex-IP problem, we have \(\bar{x} \in T_k = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1, \|x\|_1 \leq \sqrt{k}\} \subseteq \rho \cdot \text{Conv}(S_k)\). Therefore \(\bar{x}^T\bar{A}\bar{x}\) is upper bounded by the value \(\rho^2 \cdot \lambda^k(A)\). For the
first term in (18), since our SOS-2 construction enforces $g_i^2 + \frac{\rho^2}{4N^2} \geq \xi_i \geq g_i^2$, we obtain:

$$
\text{OPT}_{\text{convex-IP}} = \sum_{i \in \{i : \lambda_i > \lambda\}} (\lambda_i - \lambda)(\xi_i - \bar{g}_i^2) + \left(\lambda + \sum_{i=1}^{n}(\lambda_i - \lambda)\bar{g}_i^2\right)
$$

\[ \leq \frac{1}{4N^2} \sum_{i \in \{i : \lambda_i > \lambda\}} (\lambda_i - \lambda)\theta_i^2 + \rho^2 \cdot \lambda^k(A). \]

D Proof of Proposition 4

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

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

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

$$
\rho^2 \lambda^k(\bar{A}) = \rho^2 \lambda^k(A + (\bar{A} - A))
$$

\[ \leq \rho^2 \lambda^k(A) + \rho^2 \lambda^k(\bar{A} - A) \]

\[ \leq \rho^2 \lambda^k(A) + \rho^2(\bar{\lambda} - \lambda_{\text{min}}(A)). \]

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 $\ell_1$ 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 $\ell_2$-ball $B_{\ell_2,1} \triangleq \{x : \|x\|_2 \leq 1\}$ and $\ell_1$-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 sign$(u_i^*) = \text{sign}(v_i)$ 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 $\ell_1$-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 $N = 20$. 

26
Algorithm 3 Convex-IP Method

1: Input: Sample covariance matrix $A$, cardinality constraint $k$, size of set $\{i : \lambda_i > \lambda\}$ we desire, number of one branch splitting points $N$.
2: Output: Lower and upper bound of SPCA or $\ell_1$-relax based on the choice of $\theta_i$.
3: function Convex-IP Method($A, k, I_{pos}, N$)
4: Set lower bound and warm starting point $(LB, \bar{x}) \leftarrow$ Heuristic Method($A, k, x^0$).
5: Set parameter $\lambda_{i_{pos}+1} \leq \lambda \leq LB$ if possible, otherwise set $\lambda \leftarrow LB$.
6: Set splitting points $\gamma^j_i$ as above based on $N$ and the choice of $\theta_i$, see Section 2.2 [3].
7: To warm start, add additional splitting points based on the point $\bar{x}$.
8: Add cutting-plane (9) to the model based on the choice of $\theta_i$.
9: Run Convex-IP problem.
10: Set UB $\leftarrow$ Convex-IP if running to the optimal, or the current dual bound obtained from Convex-IP.
11: return LB, UB.
12: end function

Algorithm 4 Pert-Convex-IP Method

1: 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.
2: Output: Lower and upper bound of SPCA or $\ell_1$-relax based on the choice of $\theta_i$.
3: function Pert-Convex-IP Method($A, k, I_{pos}, N, iter$)
4: Set lower bound and warm starting point $(LB, \bar{x}) \leftarrow$ Heuristic Method($A, k, x^0$).
5: Set parameter $\lambda_{i_{pos}+1} \leq \lambda \leq LB$ if possible, otherwise set $\lambda \leftarrow LB$.
6: Set parameter $\bar{\lambda} \triangleq \max\{\lambda_i : \lambda_i \leq \lambda\} < \lambda$ if possible.
7: Set splitting points $\gamma^j_i$ as above based on $N$ and the choice of $\theta_i$, see Section 2.2 [3].
8: To warm start, add additional splitting points based on the point $\bar{x}$.
9: while current iteration does not exceed the maximum number of iterations iter or time limit is not up do
11: Set UB $\leftarrow$ Pert-Convex-IP if running to the optimal, or the current dual bound obtained from Pert-Convex-IP.
12: Set $\hat{x} \leftarrow$ current feasible solution obtained from Pert-Convex-IP
13: Add additional splitting points based on solution obtained in solving Pert-Convex-IP problem.
14: Add cutting-plane (9) to the model based on the choice of $\theta_i$.
15: end while
16: return LB, UB.
17: 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 $\ell_1$-relaxation heuristic

1: *Input:* Sample covariance matrix $A$, cardinality constraint $k$, initial feasible vector $x$.

2: *Output:* A feasible solution $x$ of $\ell_1$-relax and its objective value.

3: **function** $\ell_1$-relaxation heuristic($A, k, x$)
4: Start from a vector $x$ such that $\|x\|_2 \leq 1$, $\|x\|_1 \leq \sqrt{k}$.
5: Set current solution be $x^{\text{current}} \leftarrow x$, and past solution be $x^{\text{past}} \leftarrow 0$.
6: Set current objective value be $\text{OPT}^{\text{current}} \leftarrow (x^{\text{current}})^\top Ax^{\text{current}}$, and past objective value be $\text{OPT}^{\text{past}} \leftarrow (x^{\text{past}})^\top Ax^{\text{past}}$.
7: Let $\text{iter}$ denotes the index of current iteration, set $N$ be the maximum number of iterations, set $\epsilon$ be a stopping criteria.
8: while $\text{OPT}^{\text{current}} > \text{OPT}^{\text{past}}$ and $\|x^{\text{current}} - x^{\text{past}}\|_2 > \epsilon$ and $\text{iter} \leq N$ do
9: Set $y \leftarrow Ax^{\text{current}}$.
10: Denote the projection of $y$ onto the set $P = \{y : \|y\|_2 \leq 1, \|y\|_1 \leq \sqrt{k}\}$ as $\text{Proj}_P(y)$.
11: Set $x^{\text{past}} \leftarrow x^{\text{current}}$, and $x^{\text{current}} \leftarrow \text{Proj}_P(y)$.
12: Set $\text{OPT}^{\text{current}} \leftarrow (x^{\text{current}})^\top Ax^{\text{current}}$, and $\text{OPT}^{\text{past}} \leftarrow (x^{\text{past}})^\top Ax^{\text{past}}$.
13: end while
14: return $x^{\text{current}}, \text{OPT}^{\text{current}}$.
15: **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 $\Sigma$, 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 $\Sigma$ be:

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

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 $\Sigma \leftarrow 399 \cdot v_1v_1^\top + 299 \cdot v_2v_2^\top + I$.

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 $\hat{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} x_i x_i^\top$. 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.

28
G.1.2 Synthetic Example

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

\[
\Sigma = \begin{pmatrix}
290 \cdot 1_{n_1} 1_{n_1}^\top + I_{n_1} & 0_{n_2 \times n_2} & -87 \cdot 1_{n_1} 1_{n_3}^\top \\
0_{n_2 \times n_1} & 300 \cdot 1_{n_2} 1_{n_2}^\top + I_{n_2} & 277.5 \cdot 1_{n_2} 1_{n_3}^\top \\
-87 \cdot 1_{n_3} 1_{n_1}^\top & 277.5 \cdot 1_{n_3} 1_{n_2}^\top & 582.7875 \cdot 1_{n_3} 1_{n_3}^\top + I_{n_3}
\end{pmatrix}
\]

(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 \( \Sigma \) 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 standard benchmark to evaluate algorithms for sparse PCA.

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 Google's 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