Noname manuscript No.
(will be inserted by the editor)

Row-sparse principal component analysis with guarantees

Santanu S. Dey · Marco Molinaro · Guanyi Wang

Received: date / Accepted: date

Abstract Row-sparse principal component analysis (rsPCA), also known as principal component analysis (PCA) with global support, is the problem of finding the top-r leading principal components such that all these principal components are linear combination of a subset of k variables. rsPCA is a popular dimension reduction tool in statistics that enhances interpretability compared to regular principal component analysis (PCA). Methods for solving rsPCA in the literature are either greedy heuristics (in the special case of $r = 1$) with guarantees under restrictive statistical-models, or algorithms with stationary point convergence for some regularized reformulation of rsPCA. Crucially, none of the existing computational methods can efficiently guarantee the quality of the solutions obtained by comparing against dual bounds.

In this work we first propose a convex relaxation based on operator norms that provably approximates the feasible region of rsPCA within a $O(\sqrt{\log r})$ factor. To prove this result we use a novel random sparsification procedure that uses the *Pietsch-Grothendieck factorization theorem* and may be of independent interest. We also propose a simpler relaxation that is second-order cone representable and gives a $(1 + \sqrt{r})$-approximation for the feasible region.

Using these relaxations we then propose a convex integer program that provides a dual bound for the optimal value of rsPCA. Moreover, it also has worst-case guarantees: it is within a multiplicative/additive factor of the orig-

A preliminary version of this paper was published in [44]

Santanu S. Dey
ISyE, Geogia Institute of Technology
E-mail: santanu.dey@isye.gatech.edu

Marco Molinaro
Computer Science Department, Pontifical Catholic University of Rio de Janeiro
E-mail: mmolinaro@inf.puc-rio.br

Guanyi Wang
ISyE, Geogia Institute of Technology
E-mail: gwang93@gatech.edu

inal optimal value, the multiplicative factor being $O(\log r)$ or $O(r)$ depending on the relaxation used.

Finally, our experiments demonstrate both the viability of computing these dual bounds on instance with up to 2000 attributes and also their quality compared to baselines available.

Keywords Row sparse PCA · Randomized rounding · Convex hull

1 Introduction

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

$$\underset{\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}^r}{\arg\max} \operatorname{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A} \boldsymbol{V}\right), \tag{PCA}$$

where $\operatorname{Tr}(\cdot)$ is the trace, $\boldsymbol{A} := \frac{1}{M}\boldsymbol{X}\boldsymbol{X}^\top$ is the *sample covariance matrix*, and $\boldsymbol{I}^r$ denotes the $r \times r$ identity matrix.

Principal components usually tend to be dense, that is, the principal components usually involve almost all variables. This leads to a lack of of interpretability of the results from PCA, especially in the high-dimensional setting, e.g. clinical analysis, biological gene analysis, computer vision [9,47,24]. Moreover, anecdotally the principal component analysis is also known to generate large generalization error, and therefore makes inaccurate prediction. To enhance the interpretability and reduce the generalization error, it is natural to consider alternatives to PCA where a sparsity constraint is incorporated. There are many different choices of sparsity constraint depending on the context and application.

In this paper, we consider the *row-sparse PCA* (rsPCA) problem (see, for example [42]) defined as follows: Given a sample covariance matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, a *sparsity parameter* k ($\leq d$), the task is to find out the top-r k-sparsity principal components $\boldsymbol{V} \in \mathbb{R}^{d\times r}$ ($r \leq k$),

$$\underset{\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}^r,\ \|\boldsymbol{V}\|_0 \leq k}{\arg\max} \operatorname{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A} \boldsymbol{V}\right), \tag{rsPCA}$$

where the *row-sparsity constraint* $\|\boldsymbol{V}\|_0 \leq k$ denotes that there are at most k non-zero rows in matrix $\boldsymbol{V}$, i.e., the principal components share *global support*. Let

$$\mathcal{F} := \{\boldsymbol{V} \mid \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}^r,\ \|\boldsymbol{V}\|_0 \leq k\}$$

denote the feasible region of rsPCA and let $\operatorname{opt}^{\mathcal{F}}(\boldsymbol{A})$ denote the optimal value of rsPCA for sample covariance matrix $\boldsymbol{A}$.

1.1 Literature review

There is an extensive literature on (approximately) solving the sparse PCA problem and existing approaches can be broadly classified into the following five categories.

In the first category, instead of dealing with the non-convex sparsity constraint directly, the papers [25,51,5,31,43,7,20,13] incorporate additional regularizers to the objective function to enhance the sparsity of the solution. Similar to LASSO for sparse linear regression problem, these new formulations can be optimized via alternating-minimization type algorithms. We note here that the optimization problem presented in [25] is NP-hard to solve, and there is no convergence guarantee for the alternating-minimization method given in [51]. The papers [5,31,43,7,20,13] propose their own formulations for sparse PCA problem, and show that the alternating-minimization algorithm converges to stationary (critical) points. However, the solutions obtained using the above methods cannot guarantee the row-sparsity constraint $\|\boldsymbol{V}\|_0 \leq k$. Moreover, none of these methods are able to provide worst-case guarantees.

The second category of methods work with the convex relaxations of sparsity constraint. A majority of this work is for solving rsPCA for the case where $r = 1$. The papers [16,15,50,14,28,48] directly incorporate the sparsity constraint (for $r = 1$ case) and then relax the resulting optimization problem into some convex optimization problem – usually a semi-definite programming (SDP) relaxation. However, SDPs are often difficult to scale to large instances in practice. The paper [19] proposes a framework to find dual bounds of sparse PCA problem using convex quadratic integer program for the $r = 1$ case.

A third category of papers present fixed parameter tractable exact algorithms where the fixed parameter is usually the rank of the data matrix $\boldsymbol{A}$ and r. The paper [36] proposes an exact algorithm to find the global optimal solution of rsPCA with $r = 1$ with running-time of $O(d^{\text{rank}(\boldsymbol{A})+1} \log d)$. Later the paper [3] gives a combinatorial method for sparse PCA problem with *disjoint* supports. They show that their algorithm outputs a feasible solution within $(1 - \epsilon)$-multiplicative approximation ratio in time polynomial in data dimension d and reciprocal of ϵ, but exponential in the rank of sample covariance matrix $\boldsymbol{A}$ and r. Recently [17] provides a general method for solving rsPCA exactly with computational complexity polynomial in d, but exponential in r and $\text{rank}(\boldsymbol{A})$. The paper [17] mentions that the results obtained are of theoretical nature, and these methods may not be practically implementable.

A fourth category of results is that of specialized iterative heuristic methods for finding good feasible solutions of rsPCA [38,33,26,8,4,49,36] for the $r = 1$ case. These methods do not come with worst-case guarantees. Moreover, to the best of our understanding, there is no natural way to generalize these methods for solving rsPCA when $r > 1$.

The final category of papers are those that present algorithms that perform well under the assumption of a statistical-model. Under the assumption of an underlying statistical-model, the paper [22] presents a family of estimators for rsPCA with so-called 'oracle property' via solving semidefinite relaxation

of sparse PCA. The paper [18] analyzes a covariance thresholding algorithm (first proposed by [29]) for the $r = 1$ case. They show that this algorithm correctly recovers the support with high probability for sparse parameter k within order $\sqrt{M}$, with M being the number of samples. This sample complexity, combining with the lower bounds results in [6,32], suggest that no polynomial time algorithm can do significantly better under their statistical assumptions. There are also a series of papers [42,11,45,10,30] that provide the minimax rate of estimation for sparse PCA. However, all these papers require underlying statistical models, thus do not have worst-case guarantees in the model-free case.

1.2 Our contributions

In this paper, we propose explicit convex relaxations for the sparse PCA with provable quality of their approximations.

Explicit convex relaxations of the feasible region $\mathcal{F}$. Note that the objective function of rsPCA is that of maximizing a convex function, and so at least one of the extreme points of the feasible region $\mathcal{F}$ is an optimal solution. Hence, it is important to approximate the convex hull of the feasible region well.

Our first explicit convex relaxation for $\mathcal{F}$ has constraints based on the operator norms of the matrix $\boldsymbol{V}$ and is given by

$$\mathcal{CR}1 := \left\{ \boldsymbol{V} \in \mathbb{R}^{d\times r} \;\middle|\; \|\boldsymbol{V}\|_{\text{op}} \le 1,\ \|\boldsymbol{V}\|_{2\to 1} \le \sqrt{k},\ \sum_{i=1}^{d} \|\boldsymbol{V}_i\|_2 \le \sqrt{rk} \right\}, \tag{$\mathcal{CR}1$}$$

see Section 1.3 for the required definitions. Importantly, we prove that this relaxation is within a factor of $O(\log r)$ of the convex hull of $\mathcal{F}$.

Theorem 1 *For every positive integers d, r, k such that $1 \le r \le k \le d$ the convex relaxation $\mathcal{CR}1$ satisfies*

$$\mathcal{F} \ \subseteq\ \mathcal{CR}1 \ \subseteq\ \rho_{\mathcal{CR}1} \cdot \text{conv}\,(\mathcal{F})$$

for $\rho_{\mathcal{CR}1} = 2 + \max\{6\sqrt{2\pi},\, 18\sqrt{\log 50r}\}$. In particular $\rho_{\mathcal{CR}1} = O(\sqrt{\log r})$.

Given the simplicity of the formulation of the relaxation and its provable approximation guarantee, $\mathcal{CR}1$ seems to be an important set not only for sparse PCA but for other problems with row-sparsity constraints. To prove this result we use a novel randomized matrix sparsification procedure that given a matrix $\boldsymbol{V}^*$ in $\mathcal{CR}1$ produces a row-sparse matrix $\boldsymbol{V}$ with controlled spectral norm (hence in $\mathcal{F}$) that is close to the starting point $\boldsymbol{V}^*$. The main difficulty is effectively leveraging the information provided by the simple formulation $\mathcal{CR}1$, mainly the control of the $\|\cdot\|_{2\to 1}$ norm. For that, we employ in our sparsification a row sampling procedure where the weight of the rows are given

by the *Pietsch-Grothendieck factorization theorem* [37]. We believe this idea may also find uses in other problems with row-sparsity constraints.

The main computational drawback of this relaxation is that the norm $\|\cdot\|_{2\to 1}$ is known to be NP-hard to compute [39]. Therefore we also present a simpler convex relaxation of $\mathcal{F}$ that is second-order cone representable, and hence can be efficiently optimized over using interior-point methods.

Theorem 2 *For every d, r, k positive integers such that $1 \leq r \leq k \leq d$, there is a relaxation ($\mathcal{CR}2$) that is second-order cone representable and has the guarantee*

$$\text{conv}(\mathcal{F}) \subseteq \mathcal{CR}2 \subseteq \rho_{\mathcal{CR}2} \cdot \text{conv}(\mathcal{F}),$$

where $\rho_{\mathcal{CR}2} \leq 1 + \sqrt{r}$.

This generalizes the main theoretical result in [19] for the case $r = 1$.

Convex integer programming formulation for obtaining dual bounds. While the above relaxations allow us to convexity the feasible region, notice that the objective function of rsPCA is also non-convex (i.e. maximizing a convex function), and hence do not yet give a "full relaxation" that yield tractable upper (dual) bounds for the problem. Recall that dual bounds are typically crucial for effective computational procedures for non-convex problems, which are usually solved using branch-and-bound type algorithms, as dual bound allow pruning of the solution space.

To handle the non-convex objective function, we consider the natural approach of upper bounding the the objective function by piecewise linear functions, which can be modeled using binary variables and special ordered sets (SOS-2) [46]. Used together with a convex relaxation $\mathcal{CR}1$ or $\mathcal{CR}2$ this gives a convex integer programming relaxation for rsPCA.

Interestingly, we show that this full relaxation has a provable approximation guarantee, providing a dual bound that is within a multiplicative/additive factor of the optimal value of the original problem.

Theorem 3 *Let $opt^{\mathcal{F}}$ be the optimal value of rsPCA. Then there is a convex integer program* (CIP) *using the relaxation $\mathcal{CR}1$ or $\mathcal{CR}2$ whose optimal value $\text{ub}^{\mathcal{CR}i}$ satisfies the following:*

$$opt^{\mathcal{F}} \leq \text{ub}^{\mathcal{CR}i} \leq \rho^2_{\mathcal{CR}i} \cdot opt^{\mathcal{F}} + \text{additive-term},$$

where the additive term *depends on the input matrix $\boldsymbol{A}$ and the parameters used in piecewise linear approximation of the objective function, and $\rho_{\mathcal{CR}i}$ is the approximation guarantee from Theorems 1 and 2 for the relaxation used.*

Computational experiments. In order to evaluate both its practical viability and quality, we perform computational experiments with the full relaxation provided by our proposed convex integer program. For that we use both synthetic and real data with up to 2000 features, which yield input matrices of size 2000×2000.

Obtaining good feasible solutions, required to evaluate our dual bounds, already poses a challenge given the lack of heuristics for rsPCA when $r > 1$ and the size of the instances. To mitigate this we use an optimized version of the natural greedy heuristic that looks for which rows of $\boldsymbol{V}$ should be non-zero. This optimized version greatly saves on the number of eigenvalue computations required, which is a bottleneck of the process. Solving our convex integer program for the larger instances also requires care, and we employ cutting planes and a submatrix splitting technique to speed up the computations.

The numerical results show that our convex integer program provides within a reasonable time good upper bounds that are typically significantly better than an SDP relaxation and another baseline.

We note that a preliminary version of this paper was published in [44]. The current version has many new results, in particular $\mathcal{CR}1$ and results on its strength are completely new, and the numerical experiments have also been completely revamped.

1.3 Notation

We use regular lower case letters, for example α, to denote scalars. For a positive integer n, let $[n] := \{1, \ldots, n\}$. For a set $S \subseteq \mathbb{R}^n$ and a $\rho > 0$ denote $\rho \cdot S := \{\rho x \,|\, x \in S\}$.

We use bold lower case letters, for example $\boldsymbol{a}$, to be vectors. We denote the i-th component of a vector $\boldsymbol{a}$ as $\boldsymbol{a}_i$. Given two vectors, $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$, we represent the inner product of $\boldsymbol{u}$ and $\boldsymbol{v}$ by $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$. Sometimes it will be convenient to represent the outer product of vectors using $\otimes$, i.e., given two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{a} \otimes \boldsymbol{b}$ is the matrix where $[\boldsymbol{a} \otimes \boldsymbol{b}]_{i,j} = \boldsymbol{a}_i \boldsymbol{b}_j$. We denote the unit vector in the direction of the jth coordinate as $\boldsymbol{e}^j$.

We use bold upper case letters, for example $\boldsymbol{A}$, to denote matrices. We denote the (i, j)-th component of a matrix $\boldsymbol{A}$ as $\boldsymbol{A}_{ij}$. We use $\text{supp}(\boldsymbol{A})$ to denote the support of non-zero rows of matrix $\boldsymbol{A}$. We use regular upper case letters, for example I, to denote the set of indices. Given any matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and $I \subseteq [n], J \subseteq [m]$, we denote the sub-matrix of A with rows in I and columns in J as $\boldsymbol{A}_{I,J}$. For $I \in [m]$, to simplify notation we denote the submatrix of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ corresponding to the rows with index in I as $\boldsymbol{A}_I$ (instead of $\boldsymbol{A}_{I,[n]}$). Similarly for $i \in [m]$, we denote the i^{th} row of $\boldsymbol{A}$ as $\boldsymbol{A}_i$. For $J \in [n]$ again to simplify the notation, we denote the submatrix of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ corresponding to the columns with index in J as $\boldsymbol{A}_{\star,J}$ (instead of $\boldsymbol{A}_{[m],J}$), and for $j \in [n]$, we denote the j^{th} column of $\boldsymbol{A}$ as $\boldsymbol{A}_{\star,j}$.

For a symmetric square matrix $\boldsymbol{A}$, we denote the largest eigen-value of $\boldsymbol{A}$ as $\lambda_{\max}(\boldsymbol{A})$. Given $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, two symmetric matrices, we say that $\boldsymbol{A} \preceq \boldsymbol{B}$ if $\boldsymbol{B} - \boldsymbol{A}$ is a positive semi-definite matrix. Given $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{m\times n}$, we let $\langle \boldsymbol{U}, \boldsymbol{V}\rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} \boldsymbol{U}_{ij}\boldsymbol{V}_{ij}$ to be the inner product of matrices. We use $\boldsymbol{0}^{p,q}$ to denote the matrix of size $p \times q$ with all entries equal to zero. We use $\oplus$, as a sign of direct sum of matrices, i.e., given matrices $\boldsymbol{A} \in \mathbb{R}^{p\times q}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$,

$$\boldsymbol{A} \oplus \boldsymbol{B} := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0}^{p,n} \\ \boldsymbol{0}^{m,q} & \boldsymbol{B} \end{bmatrix}.$$

The operator norm $\|\boldsymbol{A}\|_{p\to q}$ of a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is defined as

$$\|\boldsymbol{A}\|_{p\to q} := \max_{\boldsymbol{x}\in\mathbb{R}^n, \|\boldsymbol{x}\|_p=1} \|\boldsymbol{A}\boldsymbol{x}\|_q.$$

We sometimes refer $\|\boldsymbol{A}\|_{2\to 2}$ as $\|\boldsymbol{A}\|_{\text{op}}$. Note that $\|\boldsymbol{A}\|_{\text{op}}$ is the largest singular value of $\boldsymbol{A}$. The Frobenius norm of a matrix $\boldsymbol{A}$ is denoted as $\|\boldsymbol{A}\|_F$.

2 Convex relaxations of $\mathcal{F}$

2.1 Operator-norms relaxation $\mathcal{CR}1$

In the vector case, i.e. $r = 1$ case, a natural convex relaxation for $\mathcal{F}$ is to control the sparsity via the ℓ_2 and ℓ_1 norms, namely to consider the set $\{\boldsymbol{v} \in \mathbb{R}^d \mid \|\boldsymbol{v}\|_2 \le 1,\ \|\boldsymbol{v}\|_1 \le \sqrt{k}\}$ (see [19]). It is easy to see that this is indeed a relaxation in the case $r = 1$: if $\boldsymbol{v} \in \mathcal{F}$, then by definition $\langle \boldsymbol{v}, \boldsymbol{v}\rangle = 1$ and so $\|\boldsymbol{v}\|_2 = 1$, and since $\boldsymbol{v}$ is a k-sparse vector we get, using the standard ℓ_1- vs ℓ_2-norm comparison in k-dimensional space, $\|\boldsymbol{v}\|_1 \le \sqrt{k}\cdot\|\boldsymbol{v}\|_2 = \sqrt{k}$.

Here we consider the relaxation $\mathcal{CR}1$ that generalizes this idea for any r, which we now recall:

$$\mathcal{CR}1 := \left\{ \boldsymbol{V} \in \mathbb{R}^{d\times r} \;\middle|\; \|\boldsymbol{V}\|_{\text{op}} \le 1,\ \|\boldsymbol{V}\|_{2\to 1} \le \sqrt{k},\ \sum_{i=1}^{d} \|\boldsymbol{V}_i\|_2 \le \sqrt{rk} \right\}.$$

Thus we now use both the $\ell_{2\to 1}$ norm and the sum of the length of the rows of $\boldsymbol{V}$ to take the role of the ℓ_1-norm proxy for sparsity (by convexity of norms both constraints are convex). While is it not hard to see that this is a relaxation of $\mathcal{F}$, we further show that it has a provable approximation guarantee.

Remark 1 One can replace in $\mathcal{CR}1$ the constraint $\|\boldsymbol{V}\|_{op} \le 1$ by the constraint $\begin{bmatrix} \boldsymbol{I}^r & -\boldsymbol{V} \\ -\boldsymbol{V} & \boldsymbol{I}^r \end{bmatrix} \succeq \boldsymbol{0}$, which is the convex hull of the *Stiefel manifold* $\{\boldsymbol{V} \mid \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}^r\}$ [21].

For the remainder of the section we prove the approximation guarantee that $\mathcal{CR}1$ provides for the convex hull of $\mathcal{F}$, namely that

$$\mathcal{F} \;\subseteq\; \mathcal{CR}1 \;\subseteq\; \rho_{\mathcal{CR}1}\cdot \text{conv}\,(\mathcal{F})$$

for $\rho_{\mathcal{CR}1} = 2 + \max\{6\sqrt{2\pi},\, 18\sqrt{\log 50r}\}$, thus proving Theorem 1. We prove each of these inclusions separately.

2.1.1 Proof of first inclusion in Theorem 1: $\mathcal{F} \subseteq \mathcal{CR}1$

Consider a matrix $\boldsymbol{V}$ in $\mathcal{F}$; we show that it satisfies the 3 constraints of $\mathcal{CR}1$. First, observe that

$$\begin{aligned}\|\boldsymbol{V}\|_{\text{op}} = \max_{\boldsymbol{x}\in\mathbb{R}^n, \|\boldsymbol{x}\|_2=1} \|\boldsymbol{V}\boldsymbol{x}\|_2 &= \max_{\boldsymbol{x}\in\mathbb{R}^n, \|\boldsymbol{x}\|_2=1} \sqrt{\langle \boldsymbol{V}\boldsymbol{x}, \boldsymbol{V}\boldsymbol{x}\rangle} \\ &= \max_{\boldsymbol{x}\in\mathbb{R}^n, \|\boldsymbol{x}\|_2=1} \sqrt{\langle \boldsymbol{x}, \boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{x}\rangle} = 1.\end{aligned}$$

Therefore, we obtain that for $\boldsymbol{V} \in \mathcal{F}$, we have $\|\boldsymbol{V}\|_{\text{op}} \leq 1$.

For the second constraint, by definition of $\|\cdot\|_{2\to 1}$ it is equivalent to verify that $\|\boldsymbol{V}\boldsymbol{x}\|_1 \leq \sqrt{k}$ for all $\boldsymbol{x} \in \mathbb{R}^r$ such that $\|\boldsymbol{x}\|_2 \leq 1$. Since $\boldsymbol{V}$ is k-row-sparse, $\boldsymbol{V}\boldsymbol{x}$ is a k-sparse vector and hence by ℓ_1- vs ℓ_2-norm comparison in k-dim space we get $\|\boldsymbol{V}\boldsymbol{x}\|_1 \leq \sqrt{k} \cdot \|\boldsymbol{V}\boldsymbol{x}\|_2 \leq \sqrt{k}$, where the last inequality follows $\|\boldsymbol{V}\boldsymbol{x}\|_2 \leq \|\boldsymbol{V}\|_{\text{op}}$ for all $\boldsymbol{x}$ satisfying $\|\boldsymbol{x}\|_2 \leq 1$.

For the third constraint of $\mathcal{CR}1$, since $\|\boldsymbol{V}\|_{\text{op}} \leq 1$ each column of $\boldsymbol{V}$, i.e., $\boldsymbol{V}_{\star,j}$ has a 2-norm of at most 1, and since there are r columns we have:

$$r \geq \|\boldsymbol{V}\|_F^2 = \sum_{i=1}^{d} \|\boldsymbol{V}_i\|_2^2.$$

Since V is k-row-sparse, at most k of the terms in the right-hand side is non-zero. Then again applying the ℓ_1- vs ℓ_2-norm comparison in k-dim space we get

$$\sum_{i=1}^{d} \|\boldsymbol{V}_i\|_2 \;\leq\; \sqrt{k} \cdot \sqrt{\sum_i \|\boldsymbol{V}_i\|_2^2}\,.$$

Combining the displayed inequalities gives $\sum_{i=1}^{d} \|\boldsymbol{V}_i\|_2 \leq \sqrt{rk}$, and so the third constraint of $\mathcal{CR}1$ is satisfied.

2.1.2 Proof of second inclusion in Theorem 1: $\mathcal{CR}1 \subseteq \rho_{\mathcal{CR}1} \cdot \text{conv}(\mathcal{F})$

We assume that $k \geq 40$, otherwise $r \leq k < 40$ and the result follows from Theorem 2. We prove the desired inclusion by comparing the support function of these sets (Proposition C.3.3.1 of [23]), namely we show that for every matrix $\boldsymbol{C} \in \mathbb{R}^{d\times r}$

$$\max_{\boldsymbol{V}\in\mathcal{CR}1} \langle \boldsymbol{C}, \boldsymbol{V}\rangle \;\leq\; \rho_{\mathcal{CR}1} \cdot \max_{\boldsymbol{V}\in\text{conv}(\mathcal{F})} \langle \boldsymbol{C}, \boldsymbol{V}\rangle. \tag{1}$$

It will suffice to prove the following sparsification result for the optimum of the left-hand side.

Lemma 1 *Assume $k \geq 40$. Consider $\boldsymbol{C} \in \mathbb{R}^{d\times r}$ and let $\boldsymbol{V}^*$ be a matrix attaining the maximum on the left-hand side of* (1)*, namely $\boldsymbol{V}^* \in \arg\max_{\boldsymbol{V}\in\mathcal{CR}1} \langle \boldsymbol{C}, \boldsymbol{V}\rangle$. Then there is a matrix $\boldsymbol{V}$ with the following properties:*

1. *(Operator norm)* $\|\boldsymbol{V}\|_{op} \leq 1 + \max\{\sqrt{18\pi}, 6\sqrt{\log 50r}\}$
2. *(Sparsity)* $\boldsymbol{V}$ *is* k*-row-sparse, namely* $\|\boldsymbol{V}\|_0 \leq k$
3. *(Value)* $\langle \boldsymbol{C}, \boldsymbol{V} \rangle \geq \frac{1}{2} \langle \boldsymbol{C}, \boldsymbol{V}^* \rangle$.

Indeed, if we have such a matrix $\boldsymbol{V}$ then $\frac{\boldsymbol{V}}{\|\boldsymbol{V}\|_{op}}$ belongs to the sparse set $\mathcal{F}$ and has value $\langle \boldsymbol{C}, \frac{\boldsymbol{V}}{\|\boldsymbol{V}\|_{op}} \rangle \geq \frac{1}{2 \cdot (1 + \max\{\sqrt{18\pi}, 6\sqrt{\log 50r}\})} \cdot \langle \boldsymbol{C}, \boldsymbol{V}^* \rangle$, showing that (1) holds.

For the remainder of the section we prove Lemma 1. The idea is to randomly sparsify $\boldsymbol{V}^*$ while controlling for operator norm and value. A standard procedure is to sample the rows of V^* with probability proportional to their squared length (see [27] for this and other sampling methods). However these more standard methods do not seem effectively leverage the information that $\|\boldsymbol{V}^*\|_{2\to 1} \leq \sqrt{k}$.

Instead, we use a novel sampling more adapted to the $\ell_{2\to 1}$-norm based on a weighting of the rows of $\boldsymbol{V}^*$ given by the so-called *Pietsch-Grothendieck factorization* [37]. We state it in a convenient form that follows by applying Theorem 2.2 of [40] to the transpose.

Theorem 4 (Pietsch-Grothendieck factorization) *Any matrix* $\boldsymbol{V} \in \mathbb{R}^{d \times r}$ *can be factorized as* $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{T}$ *of size* $\boldsymbol{T} \in \mathbb{R}^{d \times r}$, $\boldsymbol{W} \in \mathbb{R}^{d \times d}$*, where*

- $\boldsymbol{W}$ *is a nonnegative, diagonal matrix with* $\sum_i \boldsymbol{W}_{ii}^2 = 1$
- $\|\boldsymbol{T}\|_{op} \leq \sqrt{\pi/2} \cdot \|\boldsymbol{V}\|_{2\to 1}$.

So first apply this theorem to obtain a decomposition $\boldsymbol{V}^* = \boldsymbol{W}\boldsymbol{T}$. Notice that this means the ith row of $\boldsymbol{V}^*$ is just the ith row of $\boldsymbol{T}$ multiplied by the weight $\boldsymbol{W}_{ii}$. Define the "probability"

$$p_i := \frac{k}{6}\left(\boldsymbol{W}_{ii}^2 + \frac{\|\boldsymbol{V}_i^*\|_2}{\sum_{i'} \|\boldsymbol{V}_{i'}^*\|_2}\right),$$

and the truncation $\bar{p}_i = \min\{p_i, 1\}$ to make it a bonafide probability.[1] We then randomly sparsify V^* by keeping each row i with probability $\bar{p}_i$ and normalizing it: define the random matrix $\widetilde{\boldsymbol{V}} := \widetilde{\boldsymbol{W}}\boldsymbol{T}$, where $\widetilde{\boldsymbol{W}}$ is the random diagonal matrix with

$$\widetilde{\boldsymbol{W}}_{ii} := \varepsilon_i \frac{\boldsymbol{W}_{ii}}{\bar{p}_i},$$

and ε_i (the indicator that we keep row i) takes value 1 with probability $\bar{p}_i$ and 0 with probability $1 - \bar{p}_i$ (and the ε_i's are independent). Since $\mathbb{E}\widetilde{\boldsymbol{W}} = \boldsymbol{W}$ notice this is procedure is unbiased: $\mathbb{E}\widetilde{\boldsymbol{V}} = \boldsymbol{V}^*$.

We first show that $\widetilde{V}$ satisfies each of the desired items from Lemma 1 with good probability, and then use a union bound to exhibit a matrix that proves the lemma.

[1] For some intuition: The first term parenthesis in p_i controls the variance of $\widetilde{\boldsymbol{W}}_{ii}$, which is $\mathrm{Var}(\widetilde{\boldsymbol{W}}_{ii}) \leq \frac{\boldsymbol{W}_{ii}^2}{p_i} \leq \frac{6}{k}$; the second term controls the largest size of a row of $\widetilde{\boldsymbol{V}}$, which is $\|\widetilde{\boldsymbol{V}}_i\|_2 \leq \|\frac{\boldsymbol{V}_i^*}{p_i}\|_2 \leq \frac{6}{k} \sum_{i'} \|\boldsymbol{V}_{i'}^*\|_2$, which is at most 6 because $\boldsymbol{V}^* \in \mathcal{CR}1$.

Sparsity. The number of rows $\|\widetilde{V}\|_0$ of $\widetilde{V}$ is precisely $\sum_{i=1}^{d} \varepsilon_i$, whose expectation is at most

$$\sum_{i=1}^{d} p_i \;=\; \frac{k}{6}\Big(\sum_i \boldsymbol{W}_{ii}^2 \;+\; 1\Big) \;=\; \frac{k}{3}.$$

Employing the multiplicative Chernoff bound (Lemma 3) we get

$$\Pr\Big(\|\widetilde{\boldsymbol{V}}\|_0 > k\Big) \;\le\; \left(\frac{2e}{6}\right)^k \;<\; \frac{1}{50}, \tag{2}$$

where the last inequality uses that $k \ge 40$.

Operator norm. Let I be the indices i where $p_i \le 1$ (so $\bar{p}_i = p_i$), and $I^c = [d]\backslash I$ (so $\bar{p}_i = 1$ and hence $\widetilde{\boldsymbol{V}}_i = \boldsymbol{V}_i^*$). From triangle inequality we can see that $\|\widetilde{\boldsymbol{V}}\|_{\text{op}} \le \|\widetilde{\boldsymbol{V}}_I\|_{\text{op}} + \|\widetilde{\boldsymbol{V}}_{I^c}\|_{\text{op}}$. Moreover,

$$\|\widetilde{\boldsymbol{V}}_{I^c}\|_{\text{op}} = \|\boldsymbol{V}_{I^c}^*\|_{\text{op}} \le \|\boldsymbol{V}^*\|_{\text{op}} \le 1,$$

where the first equality is because the rows of $\widetilde{\boldsymbol{V}}_{I^c}$ are exactly equal to the rows of $\boldsymbol{V}_{I^c}^*$ and the first inequality is because deleting rows cannot increase the operator norm, and the last inequality because $\boldsymbol{V}^* \in \mathcal{F}$. Combining these observations we get that $\|\widetilde{\boldsymbol{V}}\|_{op} \le \|\widetilde{\boldsymbol{V}}_I\|_{op} + 1$, and so we focus on controlling the operator norm of $\widetilde{\boldsymbol{V}}_I$. We do that by applying a concentration inequality to the largest eivengalue of the PSD matrix $(\widetilde{\boldsymbol{V}}_I)^\top \widetilde{\boldsymbol{V}}_I$; the following is Theorem 1.1 of [41] plus a simple estimate (see for example page 65 of [35]).

Theorem 5 *Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \in \mathbb{R}^{r\times r}$ be independent, random, symmetric matrices of dimension r. Assume with probability 1 each $\boldsymbol{X}_i$ is PSD and has largest eigenvalue $\lambda_{\max}(\boldsymbol{X}_i) \le R$. Then*

$$\Pr\left(\lambda_{\max}\Big(\sum_i \boldsymbol{X}_i\Big) \ge \alpha\right) < r \cdot 2^{-\alpha/R}$$

for every $\alpha \ge 6\lambda_{\max}(\mathbb{E}\sum_i \boldsymbol{X}_i)$.

First notice that indeed $(\widetilde{\boldsymbol{V}}_I)^\top \widetilde{\boldsymbol{V}}_I$ can be written as a sum of independent PSD matrices:

$$(\widetilde{\boldsymbol{V}}_I)^\top \widetilde{\boldsymbol{V}}_I = \sum_{i\in I} \widetilde{\boldsymbol{V}}_i \otimes \widetilde{\boldsymbol{V}}_i = \sum_{i\in I} \widetilde{\boldsymbol{W}}_{ii}^2 (\boldsymbol{T}_i \otimes \boldsymbol{T}_i) = \sum_{i\in I} \varepsilon_i \,\frac{\boldsymbol{W}_{ii}^2}{p_i^2} (\boldsymbol{T}_i \otimes \boldsymbol{T}_i). \tag{3}$$

To estimate the max eigenvalue of the expected matrix, $\lambda_{\max}(\mathbb{E}\,(\widetilde{\boldsymbol{V}}_I)^\top \widetilde{\boldsymbol{V}}_I)$, by definition of p_i we have $\mathbb{E}\,\varepsilon_i \frac{\boldsymbol{W}_{ii}^2}{p_i^2} \le \frac{6}{k}$ and hence

$$\mathbb{E}\,(\widetilde{\boldsymbol{V}}_I)^\top \widetilde{\boldsymbol{V}}_I \;\preceq\; \sum_{i\in I} \frac{6}{k}(\boldsymbol{T}_i \otimes \boldsymbol{T}_i) \;\preceq\; \frac{6}{k}\sum_i (\boldsymbol{T}_i \otimes \boldsymbol{T}_i) \;=\; \frac{6}{k}\,\boldsymbol{T}^\top \boldsymbol{T}.$$

By the guarantee of the Pietsch-Grothendieck factorization $\|\boldsymbol{T}\|_{\mathrm{op}} \le \sqrt{\pi/2}\,\|\boldsymbol{V}^*\|_{2\to 1}$ and since $\boldsymbol{V}^* \in \mathcal{CR}1$ we have $\|\boldsymbol{V}^*\|_{2\to 1} \le \sqrt{k}$, so applying these bounds to the previous displayed inequality gives

$$\lambda_{\max}\Big(\mathbb{E}\,(\widetilde{\boldsymbol{V}}_I)^\top \widetilde{\boldsymbol{V}}_I\Big) \;\le\; \frac{6}{k}\lambda_{\max}(\boldsymbol{T}^\top \boldsymbol{T}) \;=\; \frac{6}{k}\|\boldsymbol{T}\|_{\mathrm{op}}^2 \;\le\; 3\pi.$$

To control the R term in Theorem 5 we look at the first equation in (3) and notice that for $i \in I$

$$\begin{aligned}\lambda_{\max}\Big(\widetilde{\boldsymbol{V}}_i \otimes \widetilde{\boldsymbol{V}}_i\Big) &= \lambda_{\max}\left(\left(\frac{\varepsilon_i}{p_i}\boldsymbol{V}_i^*\right) \otimes \left(\frac{\varepsilon_i}{p_i}\boldsymbol{V}_i^*\right)\right)\\ &\le \lambda_{\max}\left(\frac{1}{p_i^2}(\boldsymbol{V}_i^* \otimes \boldsymbol{V}_i^*)\right) = \frac{\|\boldsymbol{V}_i^*\|_2^2}{p_i^2} \le \frac{36(\sum_{i'}\|\boldsymbol{V}_{i'}^*\|_2)^2}{k^2} \le 36,\end{aligned}$$

where the last inequality uses the fact $\boldsymbol{V}^* \in \mathcal{CR}1$ and hence $\sum_{i'}\|\boldsymbol{V}_{i'}^*\|_2 \le \sqrt{rk} \le k$.

Then applying Theorem 5 with $\boldsymbol{X}_i = \widetilde{\boldsymbol{V}}_i \otimes \widetilde{\boldsymbol{V}}_i$, $R = 16$ and $\alpha = \max\{6\cdot 3\pi,\, 36\log 50r\}$ we get

$$\Pr\Big(\|\widetilde{\boldsymbol{V}}_I\|_{op} \ge \sqrt{\alpha}\Big) \;=\; \Pr\Big(\lambda_{\max}((\widetilde{\boldsymbol{V}}_I)^\top \widetilde{\boldsymbol{V}}_I) \ge \alpha\Big) \;<\; \frac{1}{50}.$$

Recalling we have $\|\widetilde{\boldsymbol{V}}\|_{\mathrm{op}} \le 1 + \|\widetilde{\boldsymbol{V}}_I\|_{\mathrm{op}}$, this gives that

$$\|\widetilde{\boldsymbol{V}}\|_{\mathrm{op}} > 1 + \max\{\sqrt{18\pi},\, 6\sqrt{\log 50r}\} \qquad \text{happens with probability at most } \frac{1}{50}. \tag{4}$$

Value. We want to show that with good probability $\langle C, \widetilde{\boldsymbol{V}}\rangle \ge \frac{1}{2}\langle C, \boldsymbol{V}^*\rangle$. We use throughout the following observation: for each row i we have $\langle \boldsymbol{C}_i, \boldsymbol{V}_i^*\rangle \ge 0$, since the set $\mathcal{CR}1$ is symmetric with respect to flipping the sign of a row and V^* maximizes $\langle \boldsymbol{C}, \boldsymbol{V}^*\rangle = \sum_i \langle \boldsymbol{C}_i, \boldsymbol{V}_i^*\rangle$.

Since $\mathbb{E}\widetilde{\boldsymbol{V}} = \boldsymbol{V}^*$, we have $\mathbb{E}\langle \boldsymbol{C}_I, \widetilde{\boldsymbol{V}}_I\rangle = \langle \boldsymbol{C}_I, \boldsymbol{V}_I^*\rangle$ and

$$\begin{aligned}\mathrm{Var}(\langle \boldsymbol{C}_I, \widetilde{\boldsymbol{V}}_I\rangle) \;&=\; \sum_{i\in I}\mathrm{Var}(\langle \boldsymbol{C}_i, \widetilde{\boldsymbol{V}}_i\rangle) = \sum_{i\in I}\mathrm{Var}\Big(\tfrac{\varepsilon_i}{p_i}\langle \boldsymbol{C}_i, \boldsymbol{V}_i^*\rangle\Big) \le \sum_{i\in I}\frac{\langle \boldsymbol{C}_i, \boldsymbol{V}_i^*\rangle^2}{p_i}\\ &\le\; \frac{6\sum_{i'}\|\boldsymbol{V}_{i'}^*\|_2}{k}\cdot\sum_{i\in I}\frac{\langle \boldsymbol{C}_i, \boldsymbol{V}_i^*\rangle^2}{\|\boldsymbol{V}_i^*\|_2} \;\le\; 6\cdot\Big(\max_{i\in I}\Big\langle \boldsymbol{C}_i, \tfrac{\boldsymbol{V}_i^*}{\|\boldsymbol{V}_i^*\|_2}\Big\rangle\Big)\cdot\langle \boldsymbol{C}_I, \boldsymbol{V}_I^*\rangle,\end{aligned}$$

where the second inequality uses the definition of p_i and the last inequality uses that $\sum_{i'}\|\boldsymbol{V}_{i'}^*\|_2 \le \sqrt{rk} \le k$ (since $\boldsymbol{V}^* \in \mathcal{CR}1$). Moreover, since $\frac{\boldsymbol{V}_i^*}{\|\boldsymbol{V}_i^*\|_2}$ also belongs to $\mathcal{CR}1$, the optimality of $\boldsymbol{V}^*$ guarantees that $\langle \boldsymbol{C}_i, \frac{\boldsymbol{V}_i^*}{\|\boldsymbol{V}_i^*\|_2}\rangle \le \langle \boldsymbol{C}, \boldsymbol{V}^*\rangle$, and so we have the variance upper bound

$$\mathrm{Var}(\langle \boldsymbol{C}_I, \widetilde{\boldsymbol{V}}_I\rangle) \;\le\; 6\cdot\langle \boldsymbol{C}, \boldsymbol{V}^*\rangle^2.$$

Using the fact that $\langle \boldsymbol{C}_{I^c}, \widetilde{\boldsymbol{V}}_{I^c} \rangle = \langle \boldsymbol{C}_{I^c}, \boldsymbol{V}^*_{I^c} \rangle$ and the one-sided Chebychev inequality (Lemma 4) we get

$$\Pr\left(\langle \boldsymbol{C}, \widetilde{\boldsymbol{V}} \rangle \leq \frac{1}{2}\langle \boldsymbol{C}, \boldsymbol{V}^* \rangle\right) = \Pr\left(\langle \boldsymbol{C}_I, \widetilde{\boldsymbol{V}}_I \rangle \leq \langle \boldsymbol{C}_I, V^*_I \rangle - \frac{1}{2}\langle \boldsymbol{C}, \boldsymbol{V}^* \rangle\right) \quad (5)$$

$$\leq \frac{6}{6+\frac{1}{4}} = 1 - \frac{1}{25}. \quad (6)$$

Concluding the proof of Lemma 1. Taking a union bound over inequalities (2), (4), and (6), we see that with positive probability $\widetilde{V}$ satisfies all items from Lemma 1. This shows the existence of the desired matrix V and concludes the proof.

2.2 Second order representable relaxation $\mathcal{CR}2$

Since an optimization problem involving the semi-definite constraint $\boldsymbol{V}^\top \boldsymbol{V} \preceq \boldsymbol{I}^r$ (equivalent to $\|\boldsymbol{V}\|_{op} \leq 1$) and the $\ell_{2\to 1}$-norm constraint $\|\boldsymbol{V}\|_{2\to 1} \leq \sqrt{k}$ may be challenging to solve in practice we consider the following further relaxation involving second-order cone constraints:

$$\mathcal{CR}2 := \left\{ \boldsymbol{V} \in \mathbb{R}^{d\times r} \;\middle|\; \begin{array}{ll} \|\boldsymbol{V}_{\star,j}\|_2^2 \leq 1 & \forall j \in [r] \\ \|\boldsymbol{V}_{\star,j_1} \pm \boldsymbol{V}_{\star,j_2}\|_2^2 \leq 2 & \forall j_1 \neq j_2 \in [r] \\ \|\boldsymbol{V}_{\star,j}\|_1 \leq \sqrt{k} & \forall j \in [r] \\ \sum_{i=1}^d \|\boldsymbol{V}_i\|_2 \leq \sqrt{rk} & \end{array} \right\}. \quad (\mathcal{CR}2)$$

This set is a relaxation of $\mathcal{CR}1$ obtained by considering the constraint $\max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2\leq} \|\boldsymbol{V}\boldsymbol{x}\|_2 = \|\boldsymbol{V}\|_{op} \leq 1$ only for the vectors $\boldsymbol{x} = \boldsymbol{e}^j$ and $\boldsymbol{x} = \frac{1}{\sqrt{2}}(\boldsymbol{e}^{j_1} \pm \boldsymbol{e}^{j_2})$, and considering the constraint $\max_{\boldsymbol{x}\,|\,\|\boldsymbol{x}\|_2\leq} \|\boldsymbol{V}\boldsymbol{x}\|_1 = \|\boldsymbol{V}\|_{2\to 1} \leq \sqrt{k}$ only for the vectors $\boldsymbol{x} = \boldsymbol{e}^j$. In particular this shows that $\mathcal{CR}2$ is a relaxation of $\mathcal{CR}1$ and hence a relaxation of $\mathcal{F}$. Moreover, we now show that it still gives a guaranteed approximation to the convex hull of $\mathcal{F}$, namely that

$$\operatorname{conv}(\mathcal{F}) \subseteq \mathcal{CR}2 \subseteq (1+\sqrt{r})\operatorname{conv}(\mathcal{F}),$$

thus proving Theorem 2.

Proof (Proof of Theorem 2) We have just argued that $\mathcal{CR}2$ is a relaxation of $\mathcal{F}$ so it suffices to show the second inclusion $\mathcal{CR}2 \subseteq (1+\sqrt{r})\operatorname{conv}(\mathcal{F})$. So consider any $\boldsymbol{V} \in \mathcal{CR}2$, and we will show $\boldsymbol{V} \in (1+\sqrt{r})\operatorname{conv}(\mathcal{F})$.

Since the sets $\mathcal{F}$ and $\mathcal{CR}2$ are symmetric to row permutations, assume without loss of generality that the rows of $\boldsymbol{V}$ are sorted in non-decreasing length, namely $\|\boldsymbol{V}_1\|_2 \geq \|\boldsymbol{V}_2\|_2 \geq \ldots$. Decompose $\boldsymbol{V}$ based on its top-k largest rows, second top-k largest rows, and so on, i.e., let $m = \lceil d/k \rceil$, $\boldsymbol{V} = \boldsymbol{V}^1 + \cdots + \boldsymbol{V}^m$ with $\boldsymbol{V}^p \in \mathbb{R}^{d\times r}$ and

$$\operatorname{supp}(\boldsymbol{V}^1) = \{1,\ldots,k\} =: I^1 \;,\; \ldots \;,\; \operatorname{supp}(\boldsymbol{V}^m) = \{d-(m-1)k,\ldots,d\} =: I^m.$$

For each $p = 1, \ldots, m$ we have $\| \|\boldsymbol{V}^p / \|\boldsymbol{V}^p\|_{\text{op}} \|_0 \leq k$ and $\| \|\boldsymbol{V}^p / \|\boldsymbol{V}^p\|_{\text{op}} \|_{\text{op}} = 1$, thus $\boldsymbol{V}^p / \|\boldsymbol{V}^p\|_{\text{op}} \in \mathcal{F}$. Observe that:

$$\boldsymbol{V} = \boldsymbol{V}^1 + \cdots + \boldsymbol{V}^m = \|\boldsymbol{V}^1\|_{\text{op}} \frac{\boldsymbol{V}^1}{\|\boldsymbol{V}^1\|_{\text{op}}} + \cdots + \|\boldsymbol{V}^m\|_{\text{op}} \frac{\boldsymbol{V}^m}{\|\boldsymbol{V}^m\|_{\text{op}}} \tag{7}$$

$$\Rightarrow \frac{\boldsymbol{V}}{\sum_{p=1}^m \|\boldsymbol{V}^p\|_{\text{op}}} = \left(\frac{\|\boldsymbol{V}^1\|_{\text{op}}}{\sum_{p=1}^m \|\boldsymbol{V}^p\|_{\text{op}}} \right) \frac{\boldsymbol{V}^1}{\|\boldsymbol{V}^1\|_{\text{op}}} + \cdots + \left(\frac{\|\boldsymbol{V}^m\|_{\text{op}}}{\sum_{p=1}^m \|\boldsymbol{V}^p\|_{\text{op}}} \right) \frac{\boldsymbol{V}^m}{\|\boldsymbol{V}^m\|_{\text{op}}}$$
$$\in \text{conv}(\mathcal{F}).$$

Notice that $\|\boldsymbol{V}^1\|_{\text{op}} \leq 1$, since $\|\boldsymbol{V}\|_{\text{op}} \leq 1$ and zeroing out rows cannot increase the operator norm, and also by standard relationship between $\|\cdot\|_2$ and $\|\cdot\|_F$ we have:

$$\|\boldsymbol{V}^p\|_{\text{op}} \ \leq \ \sqrt{\sum_{i \in I^p} \|\boldsymbol{V}_i\|_2^2}.$$

Furthermore, we can bound the norm of each of these rows of $\boldsymbol{V}^p$ by the average of the rows of $\boldsymbol{V}^{p-1}$, since the rows of $\boldsymbol{V}$ are sorted in non-decreasing length. Employing these bounds we get

$$\begin{aligned}
\sum_{p=1}^m \|\boldsymbol{V}^p\|_{\text{op}} &= \|\boldsymbol{V}^1\|_{\text{op}} + \sum_{p=2}^m \|\boldsymbol{V}^p\|_{\text{op}} \\
&\leq 1 + \sum_{p=2}^m \sqrt{\left(\frac{\sum_{i \in I^{p-1}} \|\boldsymbol{V}_i\|_2}{k} \right)^2 \cdot k} \\
&= 1 + \frac{1}{\sqrt{k}} \cdot \sum_{p=2}^m \sum_{i \in I^{p-1}} \|\boldsymbol{V}_i\|_2 \\
&\leq 1 + \frac{1}{\sqrt{k}} \sum_{i=1}^d \|\boldsymbol{V}_i\|_2 \leq \ 1 + \sqrt{r}
\end{aligned} \tag{8}$$

where the final inequality holds since the constraint $\sum_{i=1}^d \|\boldsymbol{V}_i\|_2 \leq \sqrt{rk}$ is in the description of $\mathcal{CR}2$.

Combining inequalities (7) and (8) we have

$$\boldsymbol{V} \in \left(\sum_{p=1}^m \|\boldsymbol{V}^p\|_{\text{op}} \right) \cdot \text{conv}(\mathcal{F}) \subseteq (1 + \sqrt{r}) \cdot \text{conv}(\mathcal{F}).$$

concluding the proof of the theorem.

3 Convex IP formulation for obtaining dual bounds for rsPCA

Based on the results in the Section 2, we can set-up the following optimization problem:

$$\text{opt}^{\mathcal{CR}i} := \max_{\boldsymbol{V} \in \mathcal{CR}i} \text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right). \tag{CRi-Relax}$$

The following is a straightforward Corollary of Theorem 1 and Theorem 2:

Corollary 1 $\text{opt}^{\mathcal{F}} \leq \text{opt}^{\mathcal{CR}i} \leq \rho^2_{\mathcal{CR}i}\text{opt}^{\mathcal{F}}$ *for* $i \in \{1, 2\}$.

The challenge of solving CRi-Relax is that the objective function is non-concave. Indeed, for the case $r = 1$, Corollary 1 provides constant multiplicative approximation ratios to rsPCA; thus, the inapproximability results for rsPCA with $r = 1$ from [12,34] imply that solving CRi-Relax to optimality is NP-hard. Therefore we construct a further concave relaxation of the objective function.

3.1 Piecewise linear upper approximation of objective function

Let $\boldsymbol{A} = \sum_{j=1}^d \lambda_j \boldsymbol{a}_j \boldsymbol{a}_j^\top$ be the eigenvalue decomposition of sample covariance matrix $\boldsymbol{A}$ with $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$. The objective function then can be represented as a summation

$$\text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right) = \sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} (\boldsymbol{a}_j^\top \boldsymbol{v}_i)^2$$

where $\boldsymbol{v}_i$ denotes the ith column of $\boldsymbol{V}$ such that $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r)$. Set auxiliary variables $g_{ji} = \boldsymbol{a}_j^\top \boldsymbol{v}_i$ for $(j, i) \in [r] \times [d]$. Let $\boldsymbol{a}_j \in \mathbb{R}^d$ satisfy

$$|[\boldsymbol{a}_j]_{j_1}| \geq \ldots \geq |[\boldsymbol{a}_j]_{j_k}| \geq \ldots \geq |[\boldsymbol{a}_j]_{j_d}|,$$

and let

$$\theta_j = \sqrt{[\boldsymbol{a}_j]^2_{j_1} + \cdots + [\boldsymbol{a}_j]^2_{j_k}}$$

be the square root of sum of top-k largest absolute entries of $\boldsymbol{a}_j$. Since $\boldsymbol{v}_i$ is supposed to be k-sparse, it is easy to observe that g_{ji} is within the interval $[-\theta_j, \theta_j]$.

Piecewise linear approximation: To relax the non-convex objective, we can upper approximate each quadratic term g^2_{ji} by a piecewise linear function based on a new auxiliary variable ξ_{ji} via *special ordered sets type 2* (SOS-II) constraints (PLA) as follows,

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

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

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

which split the region $[-\theta_j, \theta_j]$ into $2N$ equal intervals. See Figure 1 for an example. By using PLA, we arrive at the following *convex integer programming*

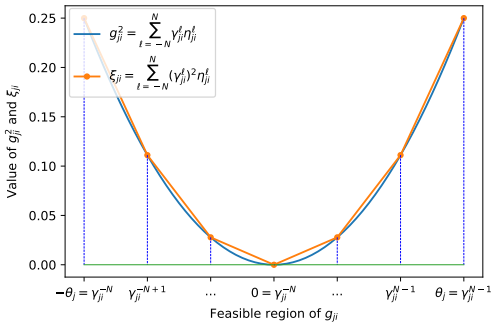

Fig. 1 The quadratic function g_{ji}^2 is upper approximated by a piecewise linear function ξ_{ji} by SOS-II constraints for all $(j,i) \in [d] \times [r]$.

problem,

$$\begin{aligned} \text{ub}^{\mathcal{CR}i} := \max \ & \textstyle\sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} \xi_{ji} \\ \text{s.t. } & \boldsymbol{V} \in \mathcal{CR}i \\ & (g, \xi, \eta) \in \text{PLA}([d] \times [r]) \end{aligned} \tag{CIP}$$

where $\mathcal{CR}i$ is the convex set defined in Section 2.1 or Section 2.2 for $i \in \{1,2\}$ respectively, and PLA is the set of constraints for piecewise-linear upper approximation of objective. Note that we say this is a convex integer program since SOS-II is modeled using binary variables.

3.2 Guarantees on the upper bounds from the convex integer program

Here we present the worst-case guarantee on the upper bound from solving convex integer program in the form of an affine function of $\text{opt}^{\mathcal{F}}$. This is a more precise restatement of Theorem 3 from the introduction.

Theorem 3 (restated) *Let $\text{opt}^{\mathcal{F}}$ be the optimal value of rsPCA. Let $\text{ub}^{\mathcal{CR}i}$ be the upper bound obtained from solving the convex integer program using $\mathcal{CR}i$*

convex relaxation of $\mathcal{F}$ for $i \in \{1,2\}$. Then:

$$opt^{\mathcal{F}} \leq \text{ub}^{\mathcal{CR}i} \leq \rho^2_{\mathcal{CR}i} \cdot opt^{\mathcal{F}} + \underbrace{\sum_{j=1}^{d} \frac{r\lambda_j \theta_j^2}{4N^2}}_{additive\ term}, \qquad \text{for } i \in \{1,2\}.$$

Proof Based on the construction for CIP, the objective function $\text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right)$ satisfies

$$\sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} (\boldsymbol{a}_j^\top \boldsymbol{v}_i)^2 = \sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} g_{ji}^2.$$

By Corollary 1, we have

$$\max_{\boldsymbol{V} \in \mathcal{CR}i} \left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right) = \max_{\boldsymbol{V} \in \mathcal{CR}i} \sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} g_{ji}^2 \leq \rho^2_{\mathcal{CR}i} \cdot \text{opt}^{\mathcal{F}},$$

for $i \in \{1,2\}$. Note that $g_{ji} \in [-\theta_j, \theta_j]$ and we have split the interval $[-\theta_j, \theta_j]$ evenly via splitting points $(\gamma_{ji}^\ell)_{\ell=-N}^{N}$ such that $\gamma_{ji}^\ell = \frac{\ell}{N} \cdot \theta_j$. For a given $j \in [d]$ and $i \in [r]$, by the definition of SOS-II sets, let $g_{ij} = \gamma_{ji}^{\ell^*} \eta_{j,i}^{\ell^*} + \gamma_{ji}^{\ell^*+1} \eta_{j,i}^{\ell^*+1}$, $\xi_{ji} = (\gamma_{ji}^{\ell^*})^2 \eta_{j,i}^{\ell^*} + (\gamma_{ji}^{\ell^*+1})^2 \eta_{j,i}^{\ell^*+1}$ and $\eta_{j,i}^{\ell^*} + \eta_{j,i}^{\ell^*+1} = 1$ for some $\ell^* \in \{-N, \ldots, N-1\}$. Thus we have:

$$\begin{aligned}
\xi_{ji} - g_{ji}^2 &= \left((\gamma_{ji}^{\ell^*})^2 \eta_{j,i}^{\ell^*} + (\gamma_{ji}^{\ell^*+1})^2 \eta_{j,i}^{\ell^*+1}\right) - \left(\gamma_{ji}^{\ell^*} \eta_{j,i}^{\ell^*} + \gamma_{ji}^{\ell^*+1} \eta_{j,i}^{\ell^*+1}\right)^2 \\
&= (\gamma_{ji}^{\ell^*})^2 \eta_{j,i}^{\ell^*} + (\gamma_{ji}^{\ell^*+1})^2 \eta_{j,i}^{\ell^*+1} - (\gamma_{ji}^{\ell^*})^2 (\eta_{j,i}^{\ell^*})^2 - (\gamma_{ji}^{\ell^*+1})^2 (\eta_{j,i}^{\ell^*+1})^2 \\
&\quad - 2\gamma_{ji}^{\ell^*} \eta_{j,i}^{\ell^*} \gamma_{ji}^{\ell^*+1} \eta_{j,i}^{\ell^*+1} \\
&= \left(\gamma_{ji}^{\ell^*+1} - \gamma_{ji}^{\ell^*}\right)^2 \eta_{ji}^{\ell^*} \eta_{ji}^{\ell^*+1} = \frac{\theta_j^2}{N^2} \eta_{ji}^{\ell^*} \eta_{ji}^{\ell^*+1} \leq \frac{\theta_j^2}{4N^2}.
\end{aligned}$$

Therefore, the objective function in CIP satisfies

$$\sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} \xi_{ji} \leq \sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} g_{ji}^2 + \sum_{j=1}^{d} \frac{r\lambda_j \theta_j^2}{4N^2} \leq \rho^2_{\mathcal{CR}i} \cdot \text{opt}^{\mathcal{F}} + \sum_{j=1}^{d} \frac{r\lambda_j \theta_j^2}{4N^2},$$

which completes the proof.

4 Greedy heuristic for rsPCA

In order to evaluate the dual bounds produced by the convex integer program from the previous section we also need good feasible solutions for rsPCA. As mentioned in the introduction, we are not aware of any heuristics for the general case $r > 1$, so in this section we describe the optimized version of the natural greedy heuristic that we will use.

We can view rsPCA as the problem

$$\max_{S\subseteq[d],|S|=k} f(S) \text{ where,}$$
$$f(S) := \left(\max_{\boldsymbol{V}\in\mathbb{R}^{d\times r} \mid \boldsymbol{V}^\top\boldsymbol{V}=\boldsymbol{I}^r, \mathrm{supp}(\boldsymbol{V})=S} \mathrm{Tr}\left(\boldsymbol{V}^\top\boldsymbol{A}\boldsymbol{V}\right)\right), \tag{9}$$

and hence solving rsPCA reduces to selecting the correct support set S. Thus, a natural algorithm is the *1-neighborhood* local search that starts with a support set S and removes/adds one index to improve the value $f(S)$. The main issue with this strategy is that it requires an expensive eigendecomposition computation for each candidate pair i/j of indices to be removed/added in order to evaluate the function f. Here we propose a much more efficient strategy that solves a proxy version of this local search move that requires only 1 eigendecomposition per round.

For that we rewrite the problem as follows. Given a sample covariance matrix $\boldsymbol{A}$, let $\boldsymbol{A}^{1/2}$ be its positive semi-definite square root such that $\boldsymbol{A} = \boldsymbol{A}^{1/2}\boldsymbol{A}^{1/2}$. Observe that $\|\boldsymbol{A}^{\frac{1}{2}} - \boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{A}^{\frac{1}{2}}\|_F^2 = \mathrm{Tr}(\boldsymbol{A}) - \mathrm{Tr}(\boldsymbol{V}^\top\boldsymbol{A}\boldsymbol{V})$, and therefore we may equivalently solve the following problem:

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

Therefore, SPCA-alt can be reformulated into a *two-stage (inner & outer) optimization problem*:

$$\min_{S\subseteq[d],\ |S|\le k} \min_{\boldsymbol{V}_S} \ \bar{f}(S, \boldsymbol{V}_S) \text{ s.t. } \ \boldsymbol{V}_S^\top\boldsymbol{V}_S = \boldsymbol{I}^r$$

where

$$\bar{f}(S, \boldsymbol{M}) := \|(\boldsymbol{A}^{1/2})_S - \boldsymbol{M}\boldsymbol{M}^\top(\boldsymbol{A}^{1/2})_S\|_F^2 + \|(\boldsymbol{A}^{1/2})_{S^C}\|_F^2 \tag{10}$$

and $S^C := [d] \setminus S$.

In order to find a solution with small $\bar{f}(S, \boldsymbol{V}_S)$ again we use a greedy swap heuristic that removes/adds one index to S. However, we avoid eigenvalue computations by keeping $\boldsymbol{M} = \boldsymbol{V}_S$ fixed and finding an improved set S' (i.e., with $\bar{f}(S', \boldsymbol{M}) \le \bar{f}(S, \boldsymbol{M})$), and only then updating the term $\boldsymbol{M}$; only the second only needs 1 eigendecomposition of $\boldsymbol{A}_{S_t,S_t}$. We describe this in more detail, letting S_t and $\boldsymbol{V}_{S_t}^t$ be the iterates at round t.

Leaving Candidate: In the t-th iteration, given the iterates S_{t-1} and $\boldsymbol{V}_{S_{t-1}}^{t-1}$ from the previous iteration, for each index $j \in S_{t-1}$, let Δ_j^{out} be

$$\Delta_j^{\text{out}} := \|\boldsymbol{A}_j^{1/2}\|_2^2 - \left\|\boldsymbol{A}_{S_{t-1}}^{1/2} - \boldsymbol{V}_{S_{t-1}}\boldsymbol{V}_{S_{t-1}}^\top\boldsymbol{A}_{S_{t-1}}^{1/2}\right\|_F^2.$$

Then let $j^{\text{out}} := \arg\min_{j\in S_{t-1}} \Delta_j^{\text{out}}$ be the candidate to leave the set S_{t-1}.

Entering Candidate: Similarly, for each $j \in S_{t-1}^C$ define Δ_j^{in} as

$$\Delta_j^{\text{in}} := \|\boldsymbol{A}_j^{1/2}\|_2^2 - \left\|(\boldsymbol{A}^{1/2})_{S_{t-1}^j} - \boldsymbol{V}_{S_{t-1}}\boldsymbol{V}_{S_{t-1}}^\top(\boldsymbol{A}^{1/2})_{S_{t-1}^j}\right\|_F^2,$$

where $S_{t-1}^j := S_{t-1} - \{j^{\text{out}}\} + \{j\}$. Then let $j^{\text{in}} := \arg\max_{j\in S_{t-1}^C} \Delta_j^{\text{in}}$.

Update Rule: If $\Delta^{\text{out}}_{j^{\text{out}}} < \Delta^{\text{in}}_{j^{\text{in}}}$ we perform the exchange with the candidates above, namely set $S_t = S_{t-1} - \{j^{\text{out}}\} + \{j^{\text{in}}\}$. In addition, we set $\boldsymbol{V}^t_{S_t}$ to be the minimizer of $\min\{f(S_t, \boldsymbol{M}) : \boldsymbol{M}^\top \boldsymbol{M} = \boldsymbol{I}^r\}$; for that we compute the eigendecomposition $\boldsymbol{A}_{S_t,S_t} = \boldsymbol{U}_{S_t}\boldsymbol{\Lambda}_{S_t}\boldsymbol{U}^\top_{S_t}$ of $\boldsymbol{A}_{S_t,S_t}$ and set $\boldsymbol{V}^t_{S_t} = (\boldsymbol{U}_{S_t})_{\star,[r]}$ to be the eigenvectors corresponding to top r eigenvalues.

If $\Delta^{\text{out}}_{j^{\text{out}}} \geq \Delta^{\text{in}}_{j^{\text{in}}}$ the algorithm stops and return the matrix $\boldsymbol{V}$ where in rows S_{t-1} equals $\boldsymbol{V}^{t-1}_{S_{t-1}}$ (i.e., $\boldsymbol{V}_{S_{t-1}} = \boldsymbol{V}^{t-1}_{S_{t-1}}$) and in rows S^C_{t-1} equals zero. The complete pseudocode is presented in Appendix B.1.

We observe that even though our procedure works only with a proxy of the original function f of the natural greedy heuristic, by construction it still finds support sets S that monotonically decrease this objective function (see Appendix B.2 for a proof).

Lemma 2 *Algorithm 1 is a monotonically decreasing algorithm with respect to the objective function f, namely $f(S_t) < f(S_{t-1})$ for every iteration t.*

5 Computational experiments

In this section we conduct computational experiments on fairly large instances to illustrate the efficiency of our proposed methods and to asses their qualities both in terms of finding good primal solutions and proving good dual bounds. We also compare our dual bound against that obtained from an SDP relaxation and from another baseline.

5.1 Methods for comparison

5.1.1 Methods for dual bounds

In order to generate dual bounds we implemented a version of our convex integer programming formulation (CIP), adding several enhancements like reduction of the number of SOS-II constraints and cutting planes in order to improve its efficiency (see [19] for related ideas for the case of $r = 1$). This implemented version is called CIP-impl, and is described in detail in Appendix C. For all experiments we use $N = 40$ as the level of discretization for the objective function in CIP-impl. (For large instances we additionally use a dimension reduction technique, which we discuss later.)

We compare our proposed dual bound with the following two baselines:

– **Baseline 1:** Sum of diagonal entries of sub-matrix:

$$\text{Baseline1} := \boldsymbol{A}_{j_1,j_1} + \cdots + \boldsymbol{A}_{j_k,j_k}, \text{ where } \boldsymbol{A}_{j_1,j_1} \geq \boldsymbol{A}_{j_2,j_3} \geq \cdots \boldsymbol{A}_{j_d,j_d}.$$

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

– **Baseline 2:** The semi-definite programming relaxation:

$$\text{SDP} := \max_{\boldsymbol{P}} \ \text{Tr}\left(\boldsymbol{A}\boldsymbol{P}\right), \ \text{s.t.} \ \ \boldsymbol{I}_d \succeq \boldsymbol{P} \succeq \boldsymbol{0}, \ \text{Tr}(\boldsymbol{P}) = r, \ \boldsymbol{1}^\top |\boldsymbol{P}| \boldsymbol{1} \leq rk.$$

Note that this is an SDP relaxation of rsPCA obtained by lifting the variables $\boldsymbol{V}$ into the product space $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^\top$.

5.1.2 Parameter for primal algorithm (lower bounds)

To obtain good feasible solutions we implemented the modified greedy neighborhood search (Algorithm 1) proposed in Section 4. For each instance we run this algorithm 400 times, where each time we pick the initial support set S_0 as a uniformly random subset of $[d]$ of size k. We allow a maximum of d iterations. The objective function value corresponding to the best solution from the 400 runs is declared as the lower bound.

5.2 Instances for numerical experiments

We conducted numerical experiments on two types of instances.

5.2.1 Artificial instances

These instances were generated artificially using ideas similar to that of the *spiked covariance matrix* [18] that have been used often to test algorithms in the $r = 1$ case. An instance **Artificial-**k^A is generated as follows.

We first choose a sparsity parameter $k^{\text{A}} \leq \frac{d}{2}$ (which will be in the range [30]) and the orthonormal vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ of dimension k^A given by

$$\boldsymbol{u}_1^\top = \left(\frac{1}{\sqrt{k^{\text{A}}}}, \ldots, \frac{1}{\sqrt{k^{\text{A}}}}\right), \quad \boldsymbol{u}_2^\top = \left(\frac{1}{\sqrt{k^{\text{A}}}}, -\frac{1}{\sqrt{k^{\text{A}}}}, \ldots, \frac{1}{\sqrt{k^{\text{A}}}}, -\frac{1}{\sqrt{k^{\text{A}}}}, \right).$$

The *block spiked covariance matrix* $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ is then computed as

$$\boldsymbol{\Sigma} := \boldsymbol{\Sigma}_1 \oplus \boldsymbol{\Sigma}_2 \oplus \boldsymbol{I}^{d-2k^{\text{A}}},$$

where $\boldsymbol{\Sigma}_1 := \ 55\boldsymbol{u}_1\boldsymbol{u}_1^\top + 52\boldsymbol{u}_2\boldsymbol{u}_2^\top \in \mathbb{R}^{k^{\text{A}} \times k^{\text{A}}}$, $\boldsymbol{\Sigma}_2 := \ 50\boldsymbol{I}_{k^{\text{A}}} \in \mathbb{R}^{k^{\text{A}} \times k^{\text{A}}}$. Finally, we sample M i.i.d. random vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M \sim N(\boldsymbol{0}_d, \boldsymbol{\Sigma})$ from the normal distribution with covariance matrix $\boldsymbol{\Sigma}$ and create the instance $\boldsymbol{A}$ as the sample covariance matrix of these vectors:

$$\boldsymbol{A} := \frac{1}{M}\left(\boldsymbol{x}_1\boldsymbol{x}_1^\top + \cdots + \boldsymbol{x}_M\boldsymbol{x}_M^\top\right).$$

In our experiments we use $d = 500$ (thus generating 500×500 matrices) and $M = 3000$ samples. Our experiments will focus on the cases $r = 2$ and $r = 3$ and we note that in these instances the optimal support set with cardinality k^{A} is different for both choices of r.

5.2.2 Real instances

The second type of instances are four real instances using the colon cancer dataset (CovColon) from [2], the lymphoma dataset (Lymph) from [1], and Reddit instances Reddit1500 and Reddit2000 from [19]. Table 1 presents the size of each instance.

name	CovColon	Lymph	Reddit1500	Reddit2000
size	500×500	500×500	1500×1500	2000×2000

Table 1 Real instances

5.3 Software & hardware

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

5.4 Performance measure

We measure the performances of CIP-impl and the baselines based on the primal-dual gap, defined as

$$\text{Gap} := \frac{\text{ub} - \text{lb}}{\text{lb}}.$$

Here $\text{ub} \in \{\text{ub}^{\text{impl}}$ ($\text{ub}^{\text{sub-mat}}$ in Section 5.6.1), Baseline-1, Baseline-2$\}$ denotes the dual bound obtained from CIP-impl or baselines. The term lb denotes the primal bound from the primal heuristic.

5.5 Numerical results for smaller instances

First we perform experiments on smaller instances of size 100×100. These instances were constructed by picking the submatrix corresponding to the top 100 largest diagonal entries from each instance listed in Section 5.2. We append a "prime" in the name of the instances to denote these smaller instances, e.g., Artificial-k^A' and CovColon'.

Time limits. We set the time limit for CIP-impl to 60 seconds and imposed no time limit on SDP. (We note that on these smaller instances SDP terminated within 600 seconds.) We also did not impose a time limit on the primal heuristic, and just note that it took less than 120 seconds on all smaller instances.

The gaps obtained by the dual bounds using CIP-impl, Baseline1, and SDP on these instances are presented in Tables 2 and 3.

name	param (r,k):	$(2,10)$	$(2,20)$	$(2,30)$	$(3,10)$	$(3,20)$	$(3,30)$
Artificial-10'	CIP-impl	**0.031**	**0.0004**	**0.0003**	**0.04**	**0.0005**	0.0004
100×100	Baseline1	3.523	4.309	4.403	2.108	2.625	2.689
	SDP	0.032	**0.0004**	**0.0003**	0.043	**0.0005**	**0.0003**
Artificial-20'	CIP-impl	0.027	**0.011**	**0.007**	**0.026**	**0.011**	**0.006**
100×100	Baseline1	3.58	7.838	8.251	2.094	4.942	5.216
	SDP	**0.02**	0.014	0.008	0.027	0.014	**0.006**
Artificial-30'	CIP-impl	0.071	0.022	**0.015**	0.074	**0.023**	**0.012**
100×100	Baseline1	3.503	7.614	11.68	2.066	4.814	7.508
	SDP	**0.03**	**0.021**	0.02	**0.051**	0.026	0.014

Table 2 Gap values for smaller artificial instances with size 100×100

name	param (r,k):	$(2,10)$	$(2,20)$	$(2,30)$	$(3,10)$	$(3,20)$	$(3,30)$
CovColon'	CIP-impl	0.12	0.119	**0.094**	0.127	0.124	0.104
100×100	Baseline1	**0.063**	**0.117**	0.132	**0.052**	**0.086**	**0.098**
	SDP	0.674	0.688	0.663	1.244	1.186	1.052
Lymp'	CIP-impl	0.329	**0.272**	**0.269**	0.225	0.296	0.32
100×100	Baseline1	**0.095**	0.277	0.392	**0.049**	**0.178**	**0.297**
	SDP	0.529	0.449	0.362	0.943	0.695	0.567
Reddit1500'	CIP-impl	**0.155**	**0.139**	**0.126**	**0.129**	**0.109**	**0.025**
100×100	Baseline1	0.695	0.396	0.99	1.197	0.811	1.294
	SDP	0.265	0.294	0.242	0.175	0.146	0.033
Reddit2000'	CIP-impl	**0.029**	**0.014**	**0.011**	**0.092**	**0.054**	**0.011**
100×100	Baseline1	0.876	1.426	1.794	0.638	1.075	1.333
	SDP	0.106	0.062	0.036	0.160	0.084	0.034

Table 3 Gap values for smaller real instances with size 100×100

Observations:

- In Table 2 we see that for the relatively easy artificial instances both CIP-impl and SDP find quite tight upper bounds.
- In Table 3 we see that for real instances SDP is substantially dominated by both CIP-impl and Baseline1.

Overall, on the 42 instances, the dual bounds from CIP-impl are best for 28 instances, the dual bounds from Baseline-1 are best for 9 instances, and the dual bounds from SDP are best for 9 instances. Since the computation of Baseline-1 scales trivially in comparison to solving the SDP, and since SDP seems to produce dual bounds of poorer quality for the more difficult real instances — in the next section we discarded SDP from the comparison.

5.6 Larger instances

5.6.1 Sub-matrix technique for largeer instances

In order to scale the convex integer program CIP-impl to handle the larger matrices, that are now up to 2000×2000, we employ the following "sub-matrix technique" to reduce the dimension.

Given a *sub-matrix ratio parameter* $m \geq 1$ satisfying $\lceil mk \rceil \leq d$, let $S := \{j_1, \ldots, j_{\lceil mk \rceil}\}$, where $\boldsymbol{A}_{j_1,j_1} \geq \cdots \geq \boldsymbol{A}_{j_{\lceil mk \rceil},j_{\lceil mk \rceil}}$, be the index set of the top-$\lceil mk \rceil$ largest diagonal entries of $\boldsymbol{A}$. Consider the blocked representation of the sample covariance matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{S,S} & \boldsymbol{A}_{S,S^C} \\ \boldsymbol{A}_{S,S^C}^\top & \boldsymbol{A}_{S^C,S^C} \end{pmatrix},$$

where $S^C := [d] \setminus S$. Then the optimal value $\text{opt}^{\mathcal{F}}$ satisfies

$$\begin{aligned} \text{opt}^{\mathcal{F}} &= \max_{\boldsymbol{V} \in \mathcal{F}} \operatorname{Tr}(\boldsymbol{V}^\top \boldsymbol{A} \boldsymbol{V}) \\ &= \max_{\boldsymbol{V} \in \mathcal{F}} \operatorname{Tr}\left((\boldsymbol{V}_S)^\top \boldsymbol{A}_{S,S} \boldsymbol{V}_S\right) + 2 \operatorname{Tr}\left((\boldsymbol{V}_S)^\top \boldsymbol{A}_{S,S^C} \boldsymbol{V}_{S^C}\right) \\ &\qquad + \operatorname{Tr}\left((\boldsymbol{V}_{S^C})^\top \boldsymbol{A}_{S^C,S^C} \boldsymbol{V}_{S^C}\right). \qquad \text{(submatrix-tech)} \end{aligned}$$

The first and third term have straight forward upper bounds. Now we need to consider the problem of finding an upper bound on $\operatorname{Tr}\left((\boldsymbol{V}_S)^\top \boldsymbol{A}_{S,S^C} \boldsymbol{V}_{S^C}\right)$.

Let S^* be the global optimal row-support set of rsPCA. Then

$$\begin{aligned} &\operatorname{Tr}\left((\boldsymbol{V}_S)^\top \boldsymbol{A}_{S,S^C} \boldsymbol{V}_{S^C}\right) \\ &= \operatorname{Tr}\left(\left((\boldsymbol{V}_{S\cap S^*})^\top \ (\boldsymbol{V}_{S\setminus S^*})^\top\right) \begin{pmatrix} \boldsymbol{A}_{S\cap S^*,S^C\cap S^*} & \boldsymbol{A}_{S\cap S^*,S^C\setminus S^*} \\ \boldsymbol{A}_{S\setminus S^*,S^C\cap S^*} & \boldsymbol{A}_{S\setminus S^*,S^C\setminus S^*} \end{pmatrix} \begin{pmatrix} \boldsymbol{V}_{S^C\cap S^*} \\ \boldsymbol{V}_{S^C\setminus S^*} \end{pmatrix} \right) \\ &= \operatorname{Tr}\left((\boldsymbol{V}_{S\cap S^*})^\top \boldsymbol{A}_{S\cap S^*,S^C\cap S^*} \boldsymbol{V}_{S^C\cap S^*}\right). \end{aligned}$$

Since $\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}^r$, then we have $\boldsymbol{V}_{S\cap S^*}^\top \boldsymbol{V}_{S\cap S^*} + \boldsymbol{V}_{S^C\cap S^*}^\top \boldsymbol{V}_{S^C\cap S^*} = \boldsymbol{I}^r$. Thus it is sufficient to consider the following optimization problem:

$$2 \max_{\boldsymbol{V}^1,\boldsymbol{V}^2} \operatorname{Tr}\left((\boldsymbol{V}^1)^\top \boldsymbol{A}_{S\cap S^*,S^C\cap S^*} \boldsymbol{V}^2\right) \text{ s.t. } (\boldsymbol{V}^1)^\top \boldsymbol{V}^1 + (\boldsymbol{V}^2)^\top \boldsymbol{V}^2 = \boldsymbol{I}^r,$$

We show in Proposition 2, proved in the appendix, that the above term is upper bounded by $\sqrt{r} \cdot \|\boldsymbol{A}_{(S\cap S^*),(S^C\cap S^*)}\|_F$.

Therefore, letting $\tilde{k} := |S \cap S^*|$ be the cardinality of the intersection, we can upper bound the right-hand side of (submatrix-tech) as

$$\text{opt}^{\mathcal{F}} \leq \text{ub}^{\text{CIP}}(\boldsymbol{A}_{S,S}; \tilde{k}) + \sqrt{r} \cdot \|\boldsymbol{A}_{S\cap S^*, S^C \cap S^*}\|_F + \text{Baseline-1}(\boldsymbol{A}_{S^C,S^C}; k - \tilde{k}),$$

where the first term $\text{ub}^{\text{CIP}}(\boldsymbol{A}_{S,S}; \tilde{k})$ is the optimal value obtained from CIP-impl with covariance matrix $\boldsymbol{A}_{S,S}$ and sparsity parameter $\tilde{k}$ (if $\tilde{k} < r$, then reset $\tilde{k} = r$), and the the third term is the value of Baseline-1 obtained from $\boldsymbol{A}_{S^C,S^C}$ with sparsity parameter $k - \tilde{k}$.

Since S^* is unknown, then the second term can be further upper bounded by

$$\|\boldsymbol{A}_{S\cap S^*, S^*\setminus S}\|_F \leq \sqrt{\left\|\boldsymbol{A}^{k-\tilde{k}}_{\{j_1\},S^C}\right\|_2^2 + \cdots + \left\|\boldsymbol{A}^{k-\tilde{k}}_{\{j_{\tilde{k}}\},S^C}\right\|_2^2} =: \text{ub}(S; \tilde{k}; S^C; k - \tilde{k}),$$

where

$$\|\boldsymbol{A}^l_{\{j\},S^C}\|_2^2 := \boldsymbol{A}^2_{j,i_1} + \cdots + \boldsymbol{A}^2_{j,i_l} \text{ with } |\boldsymbol{A}_{j,i_1}| \geq \cdots \geq |\boldsymbol{A}_{j,i_l}| \geq \ldots \text{ for all } i \in S^C,$$

and $j_1, \ldots, j_{\tilde{k}}$ are indices satisfying: $\left\|\boldsymbol{A}^{k-\tilde{k}}_{j_1,S^C}\right\|_2^2 \geq \cdots \geq \left\|\boldsymbol{A}^{k-\tilde{k}}_{j_{\tilde{k}},S^C}\right\|_2^2 \geq \cdots$.

Since $\tilde{k}$ is also not known, we arrive at our final upper bound $\text{ub}^{\text{sub-mat}}$ by considering all of its possibilities:

$$\begin{aligned}\text{opt}^{\mathcal{F}} &\leq \max_{\tilde{k}=0}^{k} \left\{\text{ub}^{\text{CIP}}(\boldsymbol{A}_{S,S}; \tilde{k}) + \sqrt{r} \cdot \text{ub}(S; \tilde{k}; S^C; k - \tilde{k}) + \text{Baseline-1}(\boldsymbol{A}_{S^C,S^C}; k - \tilde{k})\right\} \\ &=: \text{ub}^{\text{sub-mat}}.\end{aligned}$$

5.6.2 Times for larger instances

We set a more stringent time limit of 20 seconds for each CIP-impl used within the sub-matrix technique, since a number of these computations are required to compute $\text{ub}^{\text{sub-mat}}$. Again we did not set a time limit for the primal heuristic, an just note its running times as a function of the matrix size on Table 4.

size	500×500	1500×1500	2000×2000
running time	≤ 20 min	≤ 100 min	≤ 120 min

Table 4 Running time for primal heuristic

5.6.3 Results on larger instances

We compare the gap obtained by the upper bound $\text{ub}^{\text{sub-mat}}$ (CIP-impl plus sub-matrix technique) and compare it against that obtained by Baseline1 on the artificial and real instances with original sizes. These are reported on Tables 5 and 6.

On the spiked covariance matrix artificial instances we see that our dual bound $\text{ub}^{\text{sub-mat}}$ is typically orders of magnitude better than Baseline1, and is at most 0.35 for all instances. These results also illustrate that the sub-matrix ratio parameter can have a big impact on the bound obtained by the sub-matrix technique.

On the real instances, we see from Table 6 that on instances CovColon and Lymph our dual bound $\text{ub}^{\text{sub-mat}}$ performs slightly better than Baseline1 (except instance Lymph with parameters $(3, 10)$), and the gaps are overall less than 0.39. However, on instances Reddit1500 and Reddit2000 our dual bound $\text{ub}^{\text{sub-mat}}$ vastly outperforms Baseline1 on all settings of parameters. We remark that these are the largest instances in the experiments, which attest the scalability of our proposed bound.

name	param (r, k):	$(2, 10)$	$(2, 20)$	$(2, 30)$	$(3, 10)$	$(3, 20)$	$(3, 30)$
Artificial-10	$m = 1.5$	0.527	0.151	0.25	0.366	0.1	0.169
500×500	$m = 2$	0.079	0.15	0.249	0.064	0.1	0.169
	$m = 2.5$	0.079	0.15	0.248	0.064	0.099	0.168
	$m = 5$	0.071	0.145	0.241	0.056	0.099	0.293
	$m = 10$	**0.026**	**0.002**	**0.002**	**0.03**	**0.003**	**0.003**
	Baseline1	3.522	4.309	4.403	2.101	2.625	2.688
Artificial-20	$m = 1.5$	2.397	0.566	0.268	1.629	0.384	0.186
500×500	$m = 2$	0.455	0.179	0.266	0.317	0.127	0.185
	$m = 2.5$	0.606	0.178	0.265	0.463	0.126	0.184
	$m = 5$	0.097	0.176	0.261	**0.078**	0.124	0.346
	$m = 10$	**0.073**	**0.014**	**0.009**	0.139	**0.013**	**0.008**
	Baseline1	3.58	7.838	8.251	2.097	4.942	5.216
Artificial-30	$m = 1.5$	3.515	0.595	0.65	2.071	0.406	0.425
500×500	$m = 2$	3.509	0.721	0.314	2.068	0.512	0.211
	$m = 2.5$	2.304	0.709	0.312	1.586	0.511	0.209
	$m = 5$	0.474	0.225	0.305	0.365	0.158	0.468
	$m = 10$	**0.231**	**0.026**	**0.017**	**0.349**	**0.154**	**0.014**
	Baseline1	3.519	7.626	11.68	2.074	4.82	7.508

Table 5 Gap values for artificial instances.

name	param (r,k):	$(2,10)$	$(2,20)$	$(2,30)$	$(3,10)$	$(3,20)$	$(3,30)$
CovColon	$m=1.5$	0.054	0.112	0.128	0.05	0.08	0.092
500×500	$m=2$	0.051	0.107	0.126	0.062	**0.076**	0.09
	$m=2.5$	**0.05**	**0.104**	**0.124**	0.066	0.089	**0.088**
	$m=5$	0.094	0.113	0.143	0.11	0.122	2.349
	$m=10$	1.787	1.709	1.645	3.321	3.124	3.015
	Baseline1	0.063	0.118	0.133	**0.049**	0.086	0.097
Lymph	$m=1.5$	0.09	0.27	0.41	0.064	0.174	0.315
500×500	$m=2$	**0.078**	0.267	0.406	0.103	**0.171**	0.312
	$m=2.5$	0.104	**0.264**	0.403	0.155	0.194	**0.309**
	$m=5$	0.236	0.268	**0.388**	0.2	0.296	2.698
	$m=10$	2.105	1.738	1.548	4.489	3.894	3.447
	Baseline1	0.095	0.277	0.413	**0.049**	0.18	0.319
Reddit1500	$m=1.5$	0.687	0.95	0.8	0.39	0.625	0.677
1500×1500	$m=2$	0.683	0.94	0.749	0.387	0.617	0.632
	$m=2.5$	0.672	0.937	**0.727**	0.377	0.614	**0.611**
	$m=5$	0.426	**0.47**	1.068	0.346	**0.393**	1.307
	$m=10$	**0.384**	0.927	1.075	**0.316**	1.222	1.343
	Baseline1	0.695	0.962	1.199	0.396	0.635	0.848
Reddit2000	$m=1.5$	0.845	1.408	0.76	0.556	1.026	0.667
2000×2000	$m=2$	0.837	1.4	0.664	0.549	1.019	0.585
	$m=2.5$	0.827	1.396	**0.601**	0.541	1.016	**0.538**
	$m=5$	0.456	**0.436**	1.52	0.395	**0.381**	1.311
	$m=10$	**0.298**	0.866	2.234	**0.266**	1.289	1.41
	Baseline1	0.876	1.426	1.775	0.582	1.041	1.326

Table 6 Gap values for real instances.

6 Conclusion

In this paper, we proposed a scheme for producing good primal feasible solutions and dual bounds for rsPCA problem. The primal feasible solution is obtained from a monotonically improving heuristic for rsPCA problem. We showed that the solution produced by this algorithm are of very high quality by comparing the objective value of the solutions generated to upper bounds. These upper bounds are obtained using second order cone IP relaxation designed in this paper. We also presented theoretical guarantees (affine guarantee) on the quality of the upper bounds produced by the second order cone IP. The running-time for both the primal algorithm and the dual bounding heuristic are very reasonable (less than 2 hours for the 500×500 instances and less than 3.5 hours for the 2000×2000 instance). These problems are quite challenging and on some instances, we still need more techniques to close the gap. However, to the best of our knowledge, there is no comparable theoretical or computational results for solving model-free rsPCA.

References

1. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., et al.: Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature **403**(6769), 503 (2000)
2. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences **96**(12), 6745–6750 (1999)
3. Asteris, M., Papailiopoulos, D., Kyrillidis, A., Dimakis, A.G.: Sparse PCA via bipartite matchings. In: Advances in Neural Information Processing Systems, pp. 766–774 (2015)
4. Asteris, M., Papailiopoulos, D.S., Karystinos, G.N.: Sparse principal component of a rank-deficient matrix. In: 2011 IEEE International Symposium on Information Theory Proceedings, pp. 673–677. IEEE (2011)
5. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. Mathematics of Operations Research **35**(2), 438–457 (2010)
6. Berthet, Q., Rigollet, P.: Computational lower bounds for sparse pca. arXiv preprint arXiv:1304.0828 (2013)
7. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming **146**(1-2), 459–494 (2014)
8. Boutsidis, C., Drineas, P., Magdon-Ismail, M.: Sparse features for PCA-like linear regression. In: Advances in Neural Information Processing Systems, pp. 2285–2293 (2011)
9. Burgel, P.R., Paillasseur, J., Caillaud, D., Tillie-Leblond, I., Chanez, P., Escamilla, R., Perez, T., Carré, P., Roche, N., et al.: Clinical COPD phenotypes: a novel approach using principal component and cluster analyses. European Respiratory Journal **36**(3), 531–539 (2010)
10. Cai, T., Ma, Z., Wu, Y.: Optimal estimation and rank detection for sparse spiked covariance matrices. Probability theory and related fields **161**(3-4), 781–815 (2015)
11. Cai, T.T., Ma, Z., Wu, Y., et al.: Sparse PCA: Optimal rates and adaptive estimation. The Annals of Statistics **41**(6), 3074–3110 (2013)
12. Chan, S.O., Papailliopoulos, D., Rubinstein, A.: On the approximability of sparse PCA. In: Conference on Learning Theory, pp. 623–646 (2016)
13. Chen, S., Ma, S., Xue, L., Zou, H.: An alternating manifold proximal gradient method for sparse PCA and sparse CCA. arXiv preprint arXiv:1903.11576 (2019)
14. d'Aspremont, A., Bach, F., El Ghaoui, L.: Approximation bounds for sparse principal component analysis. Mathematical Programming **148**(1-2), 89–110 (2014)
15. d'Aspremont, A., Bach, F., Ghaoui, L.E.: Optimal solutions for sparse principal component analysis. Journal of Machine Learning Research **9**(Jul), 1269–1294 (2008)
16. d'Aspremont, A., Ghaoui, L.E., Jordan, M.I., Lanckriet, G.R.: A direct formulation for sparse PCA using semidefinite programming. In: Advances in neural information processing systems, pp. 41–48 (2005)
17. Del Pia, A.: Sparse PCA on fixed-rank matrices. http://www.optimization-online.org/DB_HTML/2019/07/7307.html (2019)
18. Deshpande, Y., Montanari, A.: Sparse PCA via covariance thresholding. The Journal of Machine Learning Research **17**(1), 4913–4953 (2016)
19. Dey, S.S., Mazumder, R., Wang, G.: A convex integer programming approach for optimal sparse pca. arXiv preprint arXiv:1810.09062 (2018)
20. Erichson, N.B., Zheng, P., Manohar, K., Brunton, S.L., Kutz, J.N., Aravkin, A.Y.: Sparse principal component analysis via variable projection. arXiv preprint arXiv:1804.00341 (2018)
21. Gallivan, K.A., Absil, P.: Note on the convex hull of the stiefel manifold. Technical note (2010)
22. Gu, Q., Wang, Z., Liu, H.: Sparse PCA with oracle property. In: Advances in neural information processing systems, pp. 1529–1537 (2014)
23. Hiriart-Urruty, J.B., Lemaréchal, C.: Fundamentals of convex analysis. Springer Science & Business Media (2012)

24. Jolliffe, I.T., Cadima, J.: Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **374**(2065), 20150202 (2016)
25. Jolliffe, I.T., Trendafilov, N.T., Uddin, M.: A modified principal component technique based on the LASSO. Journal of computational and Graphical Statistics **12**(3), 531–547 (2003)
26. Journée, M., Nesterov, Y., Richtárik, P., Sepulchre, R.: Generalized power method for sparse principal component analysis. Journal of Machine Learning Research **11**(Feb), 517–553 (2010)
27. Kannan, R., Vempala, S.: Randomized algorithms in numerical linear algebra. Acta Numerica **26**, 95 (2017)
28. Kim, J., Tawarmalani, M., Richard, J.P.P.: Convexification of permutation-invariant sets and applications. arXiv preprint arXiv:1910.02573 (2019)
29. Krauthgamer, R., Nadler, B., Vilenchik, D., et al.: Do semidefinite relaxations solve sparse PCA up to the information limit? The Annals of Statistics **43**(3), 1300–1322 (2015)
30. Lei, J., Vu, V.Q., et al.: Sparsistency and agnostic inference in sparse PCA. The Annals of Statistics **43**(1), 299–322 (2015)
31. Ma, S.: Alternating direction method of multipliers for sparse principal component analysis. Journal of the Operations Research Society of China **1**(2), 253–274 (2013)
32. Ma, T., Wigderson, A.: Sum-of-squares lower bounds for sparse pca. In: Advances in Neural Information Processing Systems, pp. 1612–1620 (2015)
33. Mackey, L.W.: Deflation methods for sparse PCA. In: Advances in neural information processing systems, pp. 1017–1024 (2009)
34. Magdon-Ismail, M.: NP-hardness and inapproximability of sparse PCA. Information Processing Letters **126**, 35–38 (2017)
35. Mitzenmacher, M., Upfal, E.: Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis. Cambridge university press (2017)
36. Papailiopoulos, D., Dimakis, A., Korokythakis, S.: Sparse PCA through low-rank approximations. In: International Conference on Machine Learning, pp. 747–755 (2013)
37. Pietsch, A.: Operator ideals, vol. 16. Deutscher Verlag der Wissenschaften (1978)
38. Sigg, C.D., Buhmann, J.M.: Expectation-maximization for sparse and non-negative PCA. In: Proceedings of the 25th international conference on Machine learning, pp. 960–967. ACM (2008)
39. Steinberg, D.: Computation of matrix norms with applications to robust optimization. Research thesis, Technion-Israel University of Technology **2** (2005)
40. Tropp, J.A.: Column subset selection, matrix factorization, and eigenvalue optimization. In: Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms, pp. 978–986. SIAM (2009)
41. Tropp, J.A.: User-friendly tail bounds for sums of random matrices. Foundations of computational mathematics **12**(4), 389–434 (2012)
42. Vu, V., Lei, J.: Minimax rates of estimation for sparse PCA in high dimensions. In: Artificial intelligence and statistics, pp. 1278–1286 (2012)
43. Vu, V.Q., Cho, J., Lei, J., Rohe, K.: Fantope projection and selection: A near-optimal convex relaxation of sparse PCA. In: Advances in neural information processing systems, pp. 2670–2678 (2013)
44. Wang, G., Dey, S.: Upper bounds for model-free row-sparse principal component analysis. In: Proceedings of the International Conference on Machine Learning (2020)
45. Wang, Z., Lu, H., Liu, H.: Tighten after relax: Minimax-optimal sparse PCA in polynomial time. In: Advances in neural information processing systems, pp. 3383–3391 (2014)
46. Wolsey, L.A., Nemhauser, G.L.: Integer and combinatorial optimization, vol. 55. John Wiley & Sons (1999)
47. Yeung, K.Y., Ruzzo, W.L.: Principal component analysis for clustering gene expression data. Bioinformatics **17**(9), 763–774 (2001)
48. Yongchun Li, W.X.: Exact and approximation algorithms for sparse PCA. http://www.optimization-online.org/DB_HTML/2020/05/7802.html (2020)
49. Yuan, X.T., Zhang, T.: Truncated power method for sparse eigenvalue problems. Journal of Machine Learning Research **14**(Apr), 899–925 (2013)

50. Zhang, Y., d'Aspremont, A., El Ghaoui, L.: Sparse PCA: Convex relaxations, algorithms and applications. In: Handbook on Semidefinite, Conic and Polynomial Optimization, pp. 915–940. Springer (2012)
51. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. Journal of computational and graphical statistics **15**(2), 265–286 (2006)

Appendix

A Additional concentration inequalities

We need the standard multiplicative Chernoff bound (see Theorem 4.4 [35]).

Lemma 3 (Chernoff Bound) *Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0,1]$. Then for any $\delta > 0$ we have*

$$\Pr\left(\sum_i X_i > (1+\delta)\mu\right) < \left(\frac{e}{1+\delta}\right)^{(1+\delta)\mu},$$

where $\mu = \mathbb{E}\sum_i X_i$.

We also need the one-sided Chebychev inequality, see for example Exercise 3.18 of [35].

Lemma 4 (One-sided Chebychev) *For any random variable X with finite first and second moments*

$$\Pr\left(X \leq \mathbb{E}X - t\right) \leq \frac{Var(X)}{Var(X) + t^2}.$$

B Greedy heuristic for rsPCA

B.1 Complete pseudocode

B.2 Proof of Lemma 2

By optimality of $\boldsymbol{V}_{S_t}^t$ we can see that $f(S_t) = f(S_t, \boldsymbol{V}_{S_t}^t)$ for all t. Thus, letting $\boldsymbol{G}_t := \boldsymbol{I}^k - \boldsymbol{V}_{S_t}^t (\boldsymbol{V}_{S_t}^t)^\top$ to simplify the notation, we have

$$\begin{aligned}
f(S_{t-1}) = f(S_{t-1}, \boldsymbol{V}_{S_{t-1}}^{t-1}) &= \left\|\boldsymbol{G}_t \, (\boldsymbol{A}^{1/2})_{S_{t-1}}\right\|_F^2 + \sum_{j \in S_{t-1}^C} \left\|(\boldsymbol{A}^{1/2})_j\right\|_2^2 \\
&= \left\|\boldsymbol{G}_t \, \boldsymbol{A}_{S_t}^{1/2}\right\|_F^2 + \sum_{j \in S_t^C} \left\|\boldsymbol{A}_j^{1/2}\right\|_2^2 + \underbrace{\Delta_{j^{\text{in}}}^{\text{in}} - \Delta_{j^{\text{out}}}^{\text{out}}}_{>0} \\
&> \left\|\boldsymbol{G}_t \boldsymbol{A}_{S_t}^{1/2}\right\|_F^2 + \sum_{j \in S_t^C} \left\|\boldsymbol{A}_j^{1/2}\right\|_2^2 \\
&= f(S_t, \boldsymbol{V}_{S_t}^t) = f(S_t).
\end{aligned}$$

Algorithm 1 Modified greedy neighborhood search

Input: Covariance matrix $\boldsymbol{A}$, sparsity parameter k, number of maximum iterations T

Output: A feasible solution $\boldsymbol{V}$ for rsPCA.

Initialize with $S_0 \subseteq [d]$
Compute eigendecomposition of A_{S_0}: $\boldsymbol{A}_{S_0,S_0} = \boldsymbol{U}_{S_0}\boldsymbol{\Lambda}_{S_0}\boldsymbol{U}_{S_t}^\top$, $\boldsymbol{V}_{S_0} = (\boldsymbol{U}_{S_0})_{\star,[r]}$
for $t = 1, \ldots, T$ **do**
  Compute the leaving candidate $j^{\text{out}} := \arg\min_{j \in S_{t-1}} \Delta_j^{\text{out}}$
  Compute the entering candidate $j^{\text{in}} := \arg\max_{j \in S_{t-1}^C} \Delta_j^{\text{in}}$
  if $\Delta_{j^{\text{in}}}^{\text{in}} > \Delta_{j^{\text{out}}}^{\text{out}}$ **then**
    Set $S_t := S_{t-1} - \{j^{\text{out}}\} + \{j^{\text{in}}\}$
    Compute the eigenvalue decomposition $(\boldsymbol{A}^{1/2})_{S_t} = \boldsymbol{U}_{S_t}\boldsymbol{\Lambda}_{S_t}\boldsymbol{U}_{S_t}^\top$
    Set $\boldsymbol{V}_{S_t}^t = (\boldsymbol{U}_{S_t})_{\star,[r]}$
  else
    Return the matrix $\boldsymbol{V}$ where in rows S_{t-1} equals $\boldsymbol{V}_{S_{t-1}}^{t-1}$ (i.e., $\boldsymbol{V}_{S_{t-1}} = \boldsymbol{V}_{S_{t-1}}^{t-1}$) and in rows S_{t-1}^C equals zero
  end if
end for

C Techniques for reducing the running time of CIP

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

C.1 Threshold

The first technique is to reduce the number of SOS-II constraints. Let λ_{TH} be a threshold parameter that splits the eigenvalues $\{\lambda_j\}_{j=1}^d$ of sample covariance matrix $\boldsymbol{A}$ into two parts $J^+ = \{j : \lambda_j > \lambda_{\text{TH}}\}$ and $J^- = \{j : \lambda_j \leq \lambda_{\text{TH}}\}$. The objective function $\text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right)$ satisfies

$$\text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right) = \sum_{j \in J^+} (\lambda_j - \lambda_{\text{TH}}) \sum_{i=1}^r g_{ji}^2 + \sum_{j \in J^-} (\lambda_j - \lambda_{\text{TH}}) \sum_{i=1}^r g_{ji}^2 + \lambda_{\text{TH}} \sum_{j=1}^d \sum_{i=1}^r g_{ji}^2,$$

in which the first term is convex, the second term is concave, and the third term satisfies

$$\lambda_{\text{TH}} \sum_{j=1}^d \sum_{i=1}^r g_{ji}^2 \leq r\lambda_{\text{TH}} \qquad \text{(threshold-term)}$$

due to $\sum_{j=1}^d \sum_{i=1}^r g_{ji}^2 \leq r$. Since maximizing a concave function is equivalent to convex optimization, we replace the second term by a new auxiliary variable s and the third term by its upper bound $r\lambda_{\text{TH}}$ such that

$$\text{Tr}\left(\boldsymbol{V}^\top \boldsymbol{A}\boldsymbol{V}\right) \leq \sum_{j \in J^+} (\lambda_j - \lambda_{\text{TH}}) \sum_{i=1}^r g_{ji}^2 - s + r\lambda_{\text{TH}} \qquad \text{(threshold-tech)}$$

where

$$s \geq \sum_{j \in J^-} \underbrace{(\lambda_{\text{TH}} - \lambda_j)}_{\geq 0} \sum_{i=1}^{r} g_{ji}^2 \tag{s-var}$$

is a convex constraint. We select a value of λ_{TH} so that $|J^+| = 3$. Therefore, it is sufficient to construct a piecewise-linear upper approximation for the quadratic terms g_{ji}^2 in the first term with $j \in J^+$, i.e., constraint set $\text{PLA}([J^+] \times [r])$. We thus, greatly reduce the number of SOS-II constraints from $\mathcal{O}(d \times r)$ to $\mathcal{O}(|J^+| \times r)$, i.e. in our experiemnts to $3r$ SOS-II constraints.

C.2 Cutting planes

Similar to classical integer programming, we can incorporate additional cutting planes to improve the efficiency.

Cutting plane for sparsity: The first family of cutting-planes is obtained as follows: Since $\|\boldsymbol{V}\|_0 \leq k$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ are orthogonal, by Bessel inequality, we have

$$\sum_{i=1}^{r} g_{ji}^2 = \sum_{i=1}^{r} (\boldsymbol{a}_j^\top \boldsymbol{v}_i)^2 = \boldsymbol{a}_j^\top \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{a}_j \leq \theta_j^2, \tag{sparse-g}$$

$$\sum_{i=1}^{r} \xi_{ji} \leq \theta_j^2 \left(1 + \frac{r}{4N^2}\right). \tag{sparse-xi}$$

We call these above cuts–sparse cut since θ_j is obtained from the row sparsity parameter k.

Cutting plane from objective value: The second type of cutting plane is based on the property: for any symmetric matrix, the sum of its diagonal entries are equal to the sum of its eigenvalues. Let $\boldsymbol{A}_{j_1,j_1}, \ldots, \boldsymbol{A}_{j_k,j_k}$ be the largest k diagonal entries of the sample covariance matrix $\boldsymbol{A}$, we have

Proposition 1 *The following are valid cuts for rsPCA:*

$$\sum_{j=1}^{d} \lambda_j \sum_{i=1}^{r} g_{ji}^2 \leq \boldsymbol{A}_{j_1,j_1} + \cdots + \boldsymbol{A}_{j_k,j_k}. \tag{cut-g}$$

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

$$\begin{array}{c} \sum_{j \in J^+} (\lambda_j - \lambda_{\text{TH}}) \sum_{i=1}^{r} \xi_{ji} - s + g\lambda_{\text{TH}} \leq \boldsymbol{A}_{j_1,j_1} + \cdots + \boldsymbol{A}_{j_k,j_k} + \sum_{j \in J^+} \frac{r(\lambda_j - \phi)\theta_j^2}{4N^2} \\ g \geq \sum_{j=1}^{d} \sum_{i=1}^{r} g_{ji}^2. \end{array} \tag{cut-xi}$$

C.3 Implemented version of CIP

Thus the implemented version of CIP is

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

C.4 Submatrix technique

Proposition 2 *Let $X \in \mathbb{R}^{m\times n}$ and let θ be defined as*

$$\theta := 2\max_{\boldsymbol{V}^1\in\mathbb{R}^{m\times r},\boldsymbol{V}^2\in\mathbb{R}^{n\times r}} 2\mathrm{Tr}\left((\boldsymbol{V}^1)^\top \boldsymbol{X}\boldsymbol{V}^2\right) \text{ s.t. } (\boldsymbol{V}^1)^\top\boldsymbol{V}^1 + (\boldsymbol{V}^2)^\top\boldsymbol{V}^2 = \boldsymbol{I}^r,$$

then $\theta \le \sqrt{r}\|X\|_F$

Proof

$$\begin{aligned}
&\max_{\boldsymbol{V}^1,\boldsymbol{V}^2} 2\mathrm{Tr}\left((\boldsymbol{V}^1)^\top \boldsymbol{X}\boldsymbol{V}^2\right) \text{ s.t. } (\boldsymbol{V}^1)^\top\boldsymbol{V}^1 + (\boldsymbol{V}^2)^\top\boldsymbol{V}^2 = \boldsymbol{I}^r,\\
\Leftrightarrow\quad &\max_{\boldsymbol{V}^1,\boldsymbol{V}^2} \mathrm{Tr}\left((\boldsymbol{V}^1)^\top\ (\boldsymbol{V}^2)^\top\right)\begin{pmatrix}0 & \boldsymbol{X}\\ \boldsymbol{X}^\top & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{V}^1\\ \boldsymbol{V}^2\end{pmatrix}\Big) \text{ s.t. } (\boldsymbol{V}^1)^\top\boldsymbol{V}^1 + (\boldsymbol{V}^2)^\top\boldsymbol{V}^2 = \boldsymbol{I}^r,\\
\Leftrightarrow\quad &\max_{\boldsymbol{V}} \mathrm{Tr}\left(\boldsymbol{V}^\top\begin{pmatrix}0 & \boldsymbol{X}\\ \boldsymbol{X}^\top & 0\end{pmatrix}\boldsymbol{V}\right) \text{ s.t. } \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}^r.
\end{aligned}$$

Note that the final maximization problem is equal to

$$\begin{aligned}
&\max_{\boldsymbol{V}} \mathrm{Tr}\left(\boldsymbol{V}^\top\begin{pmatrix}0 & \boldsymbol{X}\\ \boldsymbol{X}^\top & 0\end{pmatrix}\boldsymbol{V}\right) \text{ s.t. } \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}^r\\
\le\ &\sum_{i=1}^{r}\lambda_i\left(\begin{pmatrix}0 & \boldsymbol{X}\\ \boldsymbol{X}^\top & 0\end{pmatrix}\right),
\end{aligned}$$

Next we verify that the eigenvalues of

$$\begin{pmatrix}0 & X\\ X^\top & 0\end{pmatrix}$$

are $\pm$ singular values of X: Let $X = U\Sigma W^\top$. In particular, note that:

$$\begin{aligned}
\begin{pmatrix}0 & U\Sigma W^\top\\ W\Sigma U^\top & 0\end{pmatrix}\begin{bmatrix}u_i\\ w_i\end{bmatrix} &= \begin{bmatrix}U\Sigma e_i\\ W\Sigma e_i\end{bmatrix} = \sigma_i(X)\begin{bmatrix}u_i\\ w_i\end{bmatrix}\\
\begin{pmatrix}0 & U\Sigma W^\top\\ W\Sigma U^\top & 0\end{pmatrix}\begin{bmatrix}u_i\\ -w_i\end{bmatrix} &= \begin{bmatrix}-U\Sigma e_i\\ W\Sigma e_i\end{bmatrix} = -\sigma_i(X)\begin{bmatrix}u_i\\ -w_i\end{bmatrix}.
\end{aligned}$$

Therefore, we have

$$\sum_{i=1}^{r}\lambda_i\left(\begin{pmatrix}0 & \boldsymbol{X}\\ \boldsymbol{X}^\top & 0\end{pmatrix}\right) = \sum_{i=1}^{r}\sigma_i(\boldsymbol{X}) \le \sqrt{r}\|\boldsymbol{X}\|_F.$$