
Beyond Symmetry: Best Submatrix Selection for the Sparse Truncated SVD

Yongchun Li · Weijun Xie

September 30, 2024

Abstract The truncated singular value decomposition (SVD), also known as the best low-rank matrix approximation with minimum error measured by a unitarily invariant norm, has been applied to many domains such as biology, healthcare, among others, where high-dimensional datasets are prevalent. To extract interpretable information from the high-dimensional data, sparse truncated SVD (SSVD) has been used to select a handful of rows and columns of the original matrix along with the best low-rank approximation. Different from the literature on SSVD focusing on the top singular value or compromising the sparsity for the seek of computational efficiency, this paper presents a novel SSVD formulation that can select the best submatrix precisely up to a given size to maximize its truncated Ky Fan norm. The fact that the proposed SSVD problem is NP-hard motivates us to study effective algorithms with provable performance guarantees. To do so, we first reformulate SSVD as a mixed-integer semidefinite program, which can be solved exactly for small- or medium-sized instances within a branch-and-cut algorithm framework with closed-form cuts and is extremely useful for evaluating the solution quality of approximation algorithms. We next develop three selection algorithms based on different selection criteria and two searching algorithms, greedy and local search. We prove the approximation ratios for all the approximation algorithms and show that all the ratios are tight when the number of rows or columns of the selected submatrix is no larger than half of the data matrix, i.e., our derived approximation ratios are unimprovable. Our numerical study demonstrates the high solution quality and computational efficiency of the proposed algorithms. Finally, all our analysis can be extended to row-sparse PCA.

Keywords Sparse truncated SVD · Mixed-integer semidefinite program · Branch-and-cut algorithm · Approximation algorithms · Row-sparse PCA

First Author: Yongchun Li
Affiliation: Georgia Institute of Technology, Atlanta, GA 30332
E-mail: ycli@gatech.edu

Corresponding Author: Weijun Xie
Affiliation: Georgia Institute of Technology, Atlanta, GA 30332
E-mail: wxie@gatech.edu

1 Introduction

Singular value decomposition (SVD) is a common tool for data analysis in statistics and computer science. Given a data matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, its SVD is of the form

$$(\text{SVD}) \quad \mathbf{A} := \sum_{i \in [r]} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top,$$

where r denotes the rank of \mathbf{A} , and for each $i \in [r]$, $\sigma_i \in \mathbb{R}_+$ is the i th largest singular value of \mathbf{A} , $\mathbf{u}_i \in \mathbb{R}^m$ and $\mathbf{v}_i \in \mathbb{R}^n$ are the corresponding left- and right-singular vectors, respectively. In practice, data scientists often approximate data matrix \mathbf{A} by a low-rank matrix for various purposes such as information retrieval, modeling convenience, and complexity reduction (see, e.g., review papers [16, 27]). According to the generalized Eckart-Young theorem in [16], the best rank- k approximation of matrix \mathbf{A} with $k \leq r$ that minimizes the approximation error measured by a unitarily invariant norm (e.g., Frobenius, spectral, or nuclear norm), is achieved by its “ k -truncated SVD” (i.e., $\sum_{i \in [k]} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$). As shown in [46], the k -truncated SVD of matrix \mathbf{A} admits the following equivalent mathematical programming formulation

$$\|\mathbf{A}\|_{(k)} := \max_{\mathbf{U} \in \mathbb{R}^{m \times k}, \mathbf{V} \in \mathbb{R}^{n \times k}} \left\{ \text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{V}) : \mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}_k \right\}, \quad (1)$$

where the Ky Fan k -norm $\|\cdot\|_{(k)}$ is defined as the sum of k largest singular values of a matrix. Particularly, suppose that $(\mathbf{U}^*, \mathbf{V}^*)$ is an optimal solution to problem (1) with \mathbf{u}_i^* and \mathbf{v}_i^* denoting their i th columns, respectively, for each $i \in [k]$, then we must have $\mathbf{u}_i^* = \mathbf{u}_i$, $\mathbf{v}_i^* = \mathbf{v}_i$, $(\mathbf{u}_i^*)^\top \mathbf{A} \mathbf{v}_i^* = \sigma_i$ for each $i \in [k]$. Thus, this optimal solution to problem (1) can form the k -truncated SVD of matrix \mathbf{A} as below

$$(\text{Truncated SVD}) \quad \sum_{i \in [k]} (\mathbf{u}_i^*)^\top \mathbf{A} \mathbf{v}_i^* \mathbf{u}_i^* (\mathbf{v}_i^*)^\top. \quad (2)$$

1.1 Model Formulation of Our Sparse Truncated SVD

Albeit being widely-used for data simplification, denoising, and extraction of a large-scale matrix \mathbf{A} (i.e., either m or n or both of them are large), the k -truncated SVD, as the best low-rank approximation, can have difficulties in handling the following situations: (i) for the high-dimensional data, interpretable statistical estimations often require low-rank and sparse data structures (see, e.g., [52]); (ii) low-rank data approximation also often comes with the extraction of rows and columns [13]. For example, the biclustering of microarray data seeks to identify a subset of rows (e.g., cancers) and a subset of columns (e.g., genes) of a matrix that are significantly related [15, 53] and for the defect diagnostics in multistage manufacturing, it is desirable to achieve a joint selection of crucial row and column variables representing stages and processes [18]; and (iii) the vanilla k -truncated SVD often involves many rows and columns of a matrix, resulting in uninterpretable factors that do not offer insights [37, 48]. An intuitive way of resolving these issues is finding a simultaneously sparse and low-rank approximation of matrix \mathbf{A} . That is, we propose a Sparse truncated SVD (SSVD) formulation

$$(\text{SSVD}) \quad w^* := \max_{\mathbf{U} \in \mathbb{R}^{m \times k}, \mathbf{V} \in \mathbb{R}^{n \times k}} \left\{ \text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{V}) : \mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}_k, \|\mathbf{U}\|_0 \leq s_1, \|\mathbf{V}\|_0 \leq s_2 \right\}, \quad (3)$$

where for a matrix \mathbf{X} , we let $\|\mathbf{X}\|_0$ denote the number of non-zero rows, integers $s_1 \leq m$ and $s_2 \leq n$ denote the largest numbers of non-zero rows of top left- and right- singular matrices, respectively, $k \leq \min\{s_1, s_2\}$ denotes the desirable low rank, and w^* denotes the optimal value of SSVD.

Given an optimal solution to SSVD (3), following the construction in (2), we can obtain a sparse k -truncated SVD whose left- and right- singular matrices are still orthonormal and have at most s_1 and s_2 non-zero rows, respectively. Strictly enforcing the sparsity allows one to select at most s_1 rows and s_2

columns of the original data matrix and explore the inherent patterns as well as the low-rank approximation, and thus can provide a neat approximation of the data matrix and improve the interpretability of the obtained truncated SVD. In addition, SSVD (3) can be viewed as finding the best sparse and low-rank approximation of a data matrix. For many large-scale machine learning and statistical applications in various domains such as biology, manufacturing, healthcare, and others, where high-dimensional datasets are prevalent, it is desirable to pursue simultaneously sparse and low-rank approximation for better interpretable results. For example, by combining the low-rankness and sparsity, the quality of data recovery and reconstruction in signal/image processing can be considerably improved, compared to the methods solely adopting one of them [5, 34]. Our numerical study on solar flare detection via satellite images further validates the effectiveness of the proposed sparse and low-rank approximation in our SSVD (3).

Model Interpretation. Combining the k -truncated SVD (1) and the SSVD (3), we see that the SSVD (3) can be recast as the following combinatorial optimization problem:

$$(\text{SSVD}) \quad w^* := \max_{S_1 \subseteq [m], S_2 \subseteq [n]} \{ \|\mathbf{A}_{S_1, S_2}\|_{(k)} : |S_1| \leq s_1, |S_2| \leq s_2 \}, \quad (4)$$

where for any pair of subsets $S_1 \subseteq [m]$ and $S_2 \subseteq [n]$, \mathbf{A}_{S_1, S_2} denotes the submatrix of \mathbf{A} with rows and columns indexed by S_1 and S_2 , respectively.

The combinatorial formulation (4) leads to a more intuitive explanation of SSVD from the perspective of submatrix selection, which is remarked below: (i) for a given data matrix \mathbf{A} , the objective of SSVD (4) is to select the best submatrix of size at most $s_1 \times s_2$ with the maximum Ky Fan k -norm; (ii) The Ky Fan k -norm is equivalent to the sum of all the singular values of the k -truncated SVD, which is commonly used to measure information contained by the top k singular vectors (see, e.g., [13, 50]); and (iii) compared to the vanilla k -truncated SVD, the resultant sparse rank- k matrix approximation by SSVD (4) improves the interpretability while preserving the properties of truncated SVD (e.g., extracting k largest singular values of an optimal submatrix, maintaining the orthonormality of singular vectors, etc.).

Model Appliance. The SSVD (3) is versatile and generalizes many existing models. Particularly, we will show that (i) when $s_1 = m$ and $s_2 = n$, it reduces to the vanilla k -truncated SVD (1) of matrix \mathbf{A} ; (ii) when $k = 1$, it reduces to the rank-one SSVD with L_0 norm constraints proposed by [30]; and (iii) when matrix \mathbf{A} is positive semidefinite and $s_1 = s_2$, SSVD (3) reduces to the well-known row-Sparse PCA (SPCA) in [12, 45]. Formally, SPCA considered in this paper can be defined as below.

$$(\text{SPCA}) \quad w^{\text{spca}} := \max_{\mathbf{U} \in \mathbb{R}^{n \times k}} \{ \text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{U}) : \mathbf{U}^\top \mathbf{U} = \mathbf{I}_k, \|\mathbf{U}\|_0 \leq s \}, \quad (5)$$

where matrix $\mathbf{A} \in \mathcal{S}_+^n$ is assumed to be a sample covariance matrix and thus is positive semidefinite, $k \leq s \leq n$ are positive integers, and w^{spca} denotes the optimal value of SPCA. Since SPCA (5) is a special case of SSVD (3), our developed algorithms can be adapted to all the SPCA applications.

1.2 Relevant Literature

Submatrix Selection Problems. As the combinatorial formulation (4) interprets SSVD as a task of selecting an optimal submatrix, we differ our SSVD (4) from some classic submatrix selection problems in literature in the following two aspects: selected submatrix form and objective function. Given a data matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we select a rectangular submatrix of arbitrary size $s_1 \times s_2$, while the maximum volume [8, 10] and sparse ridge regression problems [47] focus on one-dimensional vector selection (i.e., $s_1 = m$ or $s_2 = n$); and SPCA [11] and maximum entropy sampling problems [21] seek to obtain the best principal submatrix out of a positive semidefinite matrix \mathbf{A} . Moreover, different from the previous works, the optimal submatrix produced by our SSVD (4) achieves the maximum Ky Fan k -norm, a natural objective for modeling a low-rank matrix.

We next review the literature on CUR matrix approximation with pseudo-skeleton approximation as a special case, which aims to show that a low-rank matrix can be well approximated using its sub-columns and sub-rows [3]. Formally, the CUR approximation is made up of the multiplied three matrices, $\mathbf{C}\mathbf{U}\mathbf{R}$,

where $C \in \mathbb{R}^{m \times k}$, $R \in \mathbb{R}^{k \times n}$ denote the selected columns and rows of data matrix $A \in \mathbb{R}^{m \times n}$, respectively, and $U \in \mathbb{R}^{k \times k}$ is the kernel matrix. An obvious limitation of the submatrix selection indicated by the CUR approximation is that the sparsity and low-rankness are not separable, i.e., it fails to guarantee a rank- k approximation when selecting $s_1 \geq k$ rows in C and $s_2 \geq k$ columns in R , which is quite different from ours. In fact, a majority of existing research about CUR approximation focused on an ill-conditioned rank- k matrix A and attempted to select a size $k \times k$ submatrix [9, 44, 54] or larger-sized submatrix [29, 36]. In contrast, our SSVD (9) can be applied to any data matrix and selecting an arbitrary number of rows and columns to achieve a low rank- k approximation.

Other submatrix selection works have compromised to use easy-to-solve but much less accurate models by relaxing the sparsity constraints by convex ones (e.g., ℓ_1 norm in [13]). Quite differently, our proposed SSVD (3) aims to select a prespecified-sized submatrix. As far as we are concerned, our SSVD (3) is the first-known rectangular submatrix selection model, maximizing the Ky Fan k -norm. Therefore, the existing submatrix selection algorithms cannot be directly used to solve SSVD (3), and none of them admits any theoretical performance guarantees compared to the true optimal value. In the numerical study, we tailor two commonly-used methods- randomized leverage-score algorithm [26] and maximum volume-based algorithm [29], to solve SSVD (3) and compare them with our proposed ones.

SSVD Models and Methods. The concept of SSVD dates back to the pioneering work by [48], which penalized the non-zero elements of rank-one truncated SVD for improving interpretability. To better convey the advantage of the proposed SSVD formulation, we next survey the existing formulations on SSVD and explain their limitations compared to ours. First, many works have mainly focused on rank-one SSVD (i.e., $k = 1$ in SSVD (3)) and the L_1 norm constraints or regularization on the top left- and right- singular vector variables $U \in \mathbb{R}^{m \times 1}$ and $V \in \mathbb{R}^{n \times 1}$ to obtain approximately sparse top singular vectors (see [22, 32, 35, 42, 48]). Albeit being computationally efficient, using L_1 -norm may not always guarantee the numbers of non-zero rows and columns in the sparse truncated SVD to be no larger than given thresholds and thus may fail to deliver desirable interpretable results. A few works recently developed on rank-one SSVD impose the L_0 norm constraints on the top left- and right- singular vector variables $U \in \mathbb{R}^{m \times 1}$ and $V \in \mathbb{R}^{n \times 1}$, restricting the size of selected submatrix exactly to be less than or equal to a given threshold [23, 30, 31].

However, the work on the general rank- k SSVD based on L_0 norm constraints is scarce. In [22, 42, 48], to construct a sparse and rank- k matrix approximation, they suggested selecting k different L_1 norm constrained rank-one SSVD sequentially; namely, at each iteration, the authors proposed to subtract the sum of the obtained sparse rank-one matrices from the original data matrix and apply their rank-one SSVD methods to the resultant matrix. Unfortunately, their selected left- and right- singular vectors do not guarantee the orthogonality nor share the same row sparsity. Another relevant work is [52], which proposed a sparse SVD algorithm to generate k sparse and orthonormal singular vectors for a better statistical estimation; however, it failed to enforce the sparsity of the obtained rank- k matrix strictly. Different from these works, this paper studies the general rank- k SSVD (3) model by strictly enforcing the sparsity and orthonormality of the obtained left- and right- singular vectors simultaneously.

Exact and Approximation Algorithms of SSVD. We show that SSVD (3) is NP-hard with a reduction to the special rank-one SPCA, i.e., $k = 1$ in formulation (5) that has been notoriously known to be NP-hard and inapproximable (see, e.g., [25]). Specifically, a seminal work on the rank-one SPCA [4] has shown that unless $P=NP$, the rank-one SPCA (5) cannot be approximated by any polynomial-time algorithm within $(1 - \epsilon)$, where $\epsilon > 0$ is some small constant. Recently, [25] further excluded a polynomial-time $(1 - O(1/s^2))$ -approximation algorithm for the rank-one SPCA (5). This motivates us to develop efficient exact and approximation algorithms for solving SSVD. Notably, the rank-one SPCA has been extensively studied in the literature (e.g., see [2, 20, 23]). In [23], the authors also derived exact mixed-integer semidefinite program (MISDP) formulations for both rank-one SPCA and rank-one SSVD. As far as we are concerned, all the existing exact algorithms and formulations are not directly applicable to SSVD (3). To fill the gap, in this paper, we derive an exact MISDP formulation and a branch-and-cut algorithm for general rank- k SSVD (3).

For the large-scale instances, it is, in general, difficult to solve SSVD (7) or its special case SPCA to optimality within a reasonable amount of time (e.g., within an hour) even when the rank $k = 1$ (see, e.g., [2]).

To speed up the computation, another line of research on high-dimensional rank-one SPCA and rank-one SSVD has adopted effective iterative methods or approximation algorithms to find a good-quality feasible solution [4, 7, 23, 30, 31]. Unfortunately, to the best of our knowledge, there are not any known approximation algorithms that can directly solve our proposed general rank- k SSVD (4) for a possibly nonsymmetric matrix. Although some algorithms manage to yield feasible solutions for SSVD (4), they fail to guarantee a provable performance. Recently, the work [12] proposed a modified local search algorithm for the general rank- k SPCA but the authors did not show the approximation ratio and their algorithm cannot be extended to nonsymmetric rank- k SSVD. This paper proposes new selection algorithms and successfully tailors greedy and local search algorithms to solve SSVD (3) with provable approximation ratios. Our first selection algorithm matches the best-known approximation ratio (i.e., $1/\sqrt{\min\{s_1, s_2\}}$) for rank-one SSVD in [23] and for the rank- k SPCA, as a special case of SSVD (3), all the proposed approximation algorithms come with the same or better performance guarantees.

1.3 Summary of Main Contributions

To solve SSVD (3), we develop an exact branch-and-cut algorithm and effective approximation algorithms using various selection and searching criteria. Below we list the major contributions.

- (i) Based on an equivalent MISDP reformulation, we derive a family of valid inequalities and a branch-and-cut algorithm for SSVD (3), which can solve the small- and medium-sized instances to optimality (e.g., $m = n = 500$, $s_1 = s_2 = 5$, $k = 2$) and help evaluate the solution quality of our proposed approximation algorithms;
- (ii) Inspired by the combinatorial formulation (4) of SSVD, we consider three different selection criteria that are related to its objective function (i.e., the Ky Fan k -norm) yet much easier to compute, and then propose three selection algorithms;
- (iii) We successfully customize the well-known greedy and local search algorithms to solve SSVD (4);
- (iv) The approximation ratios and time complexities of our proposed approximation algorithms for SSVD (4) are displayed in Table 1. We prove that all the ratios are tight, i.e., un-improvable when the sparse parameters satisfy $s_1 \leq m/2$ and $s_2 \leq n/2$;
- (v) We remark that the approximation ratio of selection Algorithm 1 is independent of k and matches the best-known one of rank-one SPCA in literature;
- (vi) All our analyses of exact and approximation algorithms can be extended to the general SPCA (5), where the results of approximation algorithms are displayed in Table 2; and
- (vii) The numerical study on large-scale instances (e.g., $m = 16313$, $n = 2365$) shows the high solution quality and scalability of our proposed approximation algorithms.

Table 1: Summary of approximation algorithms for SSVD (4)

Algorithm	Selection Algorithms			Searching Algorithms	
	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4	Algorithm 5
Ratio	$1/\sqrt{\min\{s_1, s_2\}}$	$1/\sqrt{k \min\{s_1, s_2\}}$	$\sqrt{s_1 s_2}/(k\sqrt{mn})$	$1/\sqrt{k s_1 s_2}$	$1/\sqrt{k s_1 s_2}$
Complexity	NP-hard	$O((m+n)(m \log(m) + n \log(n) + k s_1 s_2))$	$O(m \log(m) + n \log(n) + mn)$	$O(\max\{s_1, s_2\} (m+n) k s_1 s_2)$	$O(L/\delta k s_1 s_2 (n s_1 + m s_2))$

¹ L = encoding length of \mathbf{A} , and $\delta > 0$ is the strict improvement factor

Organization. The remainder of this paper is organized as follows. Section 2 presents important preliminary results. Section 3 shows equivalent formulations for SSVD. Section 4 describes approximation algorithms of SSVD with provable performance guarantees. Section 5 extends the analysis to SPCA. Section 6 introduces the branch-and-cut algorithms for exactly solving SPCA and SSVD. Section 7 presents the numerical illustration of our proposed algorithms. Section 8 concludes the paper.

Table 2: Summary of approximation algorithms for SPCA (5)

	Selection Algorithms			Searching Algorithms	
Algorithm	Algorithm 6	Algorithm 7	Algorithm 8	Algorithm 9	Algorithm 10
Ratio	$1/\sqrt{s}$	$1/\sqrt{ks}$	$s/(kn)$	k/s	k/s
Complexity	NP-hard	$O(n(n \log(n) + ks^2))$	$O(n \log(n) + n^2)$	$O(nks^3)$	$O(L/\delta nks^3)$

¹ L = encoding length of \mathbf{A} , and $\delta > 0$ is the strict improvement factor

Notation. The following notation is used throughout the paper. We use bold lower-case letters (e.g., \mathbf{x}) and bold upper-case letters (e.g., \mathbf{X}) to denote vectors and matrices, respectively, and use the corresponding non-bold letters (e.g., x_i, X_{ij}) to denote their components. For positive integers n and $s \leq n$, we let $[n] := \{1, 2, \dots, n\}$, let $[s, n] = \{s, s+1, \dots, n\}$, let \mathcal{S}_+^n denote set of all the $n \times n$ symmetric positive semidefinite matrices, and let \mathbf{I}_n denote the $n \times n$ identity matrix. For any two positive integers i, j , we let $\mathbf{0}_{i,j}$ denote a size $i \times j$ all-zeros matrix and let $\mathbf{0}_i$ denote a size i all-zeros vector, and we let $\mathbf{1}_{i,j}$ to denote a size $i \times j$ all-ones matrix and let $\mathbf{1}_i$ to denote a size i all-ones vector. For a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ with rank r , let $\mathbf{X}_{:,j}$ denote its j th column for $j \in [n]$, let $\mathbf{X}_{i,:}$ denote its i th row for each $i \in [m]$, let $\sigma_i(\mathbf{X})$ denote its i th largest singular value for each $i \in [r]$, let $\|\mathbf{X}\|_2$, $\|\mathbf{X}\|_*$, and $\|\mathbf{X}\|_F$ denote the induced two norm, nuclear, and Frobenius norm, and if \mathbf{X} is symmetric, we let $\mathbf{X} \succeq 0$ denote the positive semi-definiteness of matrix \mathbf{X} and let $\lambda_i(\mathbf{X})$ denote its i th largest eigenvalue for each $i \in [r]$. For a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and any two subsets $S_1 \subseteq [m]$ and $S_2 \subseteq [n]$, let $\mathbf{X}_{S_1,j}$ denote the j th column of submatrix $\mathbf{X}_{S_1,[n]}$ for all $j \in [n]$, and let \mathbf{X}_{i,S_2} denote the i th row of submatrix $\mathbf{X}_{[m],S_2}$ for all $i \in [m]$. For a vector $\mathbf{x} \in \mathbb{R}^n$ and any subset $S \subseteq [n]$, we let $\|\mathbf{x}\|_2$ denote the two norm, let $\|\mathbf{x}\|_\infty$ denote the infinite norm, and let \mathbf{x}_S denote the subvector with entries indexed by S . Give a real number $x \in \mathbb{R}$, we let $\text{sign}(x)$ denote its sign. All vectors and matrices are properly sized so that algebraic operations can be carried through. Additional notation will be introduced later as needed.

2 Preliminaries

According to the definition of SVD, we observe that

Observation 1 For any matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$, suppose that $\mathbf{B} = \sum_{i \in [r]} d_i \mathbf{u}_i \mathbf{v}_i^\top$ where $\mathbf{d} \in \mathbb{R}_+^r$ and both $\{\mathbf{u}_i\}_{i \in [r]}$ and $\{\mathbf{v}_i\}_{i \in [r]}$ are orthogonal (but possibly not orthonormal) vectors, respectively. Then the SVD of matrix \mathbf{B} can be represented as

$$\mathbf{B} := \sum_{i \in [r]} d_i \|\mathbf{u}_i\|_2 \|\mathbf{v}_i\|_2 \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2} \frac{\mathbf{v}_i^\top}{\|\mathbf{v}_i\|_2},$$

and $\{d_i \|\mathbf{u}_i\|_2 \|\mathbf{v}_i\|_2\}_{i \in [r]}$ are the singular values of matrix \mathbf{B} , and $\{\mathbf{u}_i / \|\mathbf{u}_i\|_2\}_{i \in [r]}$ and $\{\mathbf{v}_i / \|\mathbf{v}_i\|_2\}_{i \in [r]}$ are corresponding left- and right- orthonormal singular vectors.

We use the notion of Sylvester's Hadamard matrix to construct worst-case examples to prove the tightness of the approximation ratios of our approximation algorithms.

Definition 1 (Sylvester's Hadamard Matrix) According to Sylvester's construction [39], for every positive integer t , there exists a symmetric $2^t \times 2^t$ matrix whose entries are all equal to 1 or -1 and columns are orthogonal to each other. This is known as Sylvester's Hadamard matrix of order 2^t , denoted by $\mathbf{H}(t)$.

To be specific, Sylvester's Hadamard matrix can be constructed sequentially, i.e., $\mathbf{H}(t+1) = \begin{pmatrix} \mathbf{H}(t) & \mathbf{H}(t) \\ \mathbf{H}(t) & -\mathbf{H}(t) \end{pmatrix}$ with $\mathbf{H}(1) = [1]$. For simplicity, we refer to Sylvester's Hadamard matrix as the Hadamard matrix throughout this paper.

3 Exact MISDP of SSVD

In this section, we develop an exact mixed-integer semidefinite programming (MISDP) formulation for SSVD (3). The MISDP relies on the following key lemma that establishes a semidefinite representation of the Ky Fan k -norm for a given matrix, motivated by the variational formulation of the top k singular vectors in [17] and the facts that the largest singular value and the sum of squares of k largest singular values of a matrix have been shown to be semidefinite representable in [13, 23].

Lemma 1 For any matrix $B \in \mathbb{R}^{m \times n}$ with rank r , suppose that $B = \sum_{i \in [r]} \sigma_i(B) \bar{u}_i \bar{v}_i^\top$ denotes the SVD of B . Then for a positive integer $k \leq r$,

$$\max_{X \in \mathbb{R}^{n \times m}} \{ \text{tr}(BX) : \|X\|_2 \leq 1, \|X\|_* \leq k \} = \|B\|_{(k)},$$

where the equality can be achieved by $X^* = \sum_{i \in [k]} \bar{v}_i (\bar{u}_i)^\top$.

Proof: To show the identity, let us consider the following maximization problem

$$\max_{X \in \mathbb{R}^{n \times m}} \{ \text{tr}(BX) : \|X\|_2 \leq 1, \|X\|_* \leq k \}. \quad (6)$$

First, for any feasible solution (U, V) to the maximization problem (1), we can construct a feasible solution $X = VU^\top$ to problem (6) that yields the same objective value. Thus, the optimal value of problem (6) is lower bounded by $\|B\|_{(k)}$.

On the other hand, suppose that an optimal solution X^* of problem (6) has rank $d \geq k$ and its SVD is $X^* = \sum_{i \in [d]} \sigma_i(X^*) p_i q_i^\top$. As $\|X^*\|_2 \leq 1$, we have $\sigma_i(X^*) \leq 1$ for each $i \in [d]$. Let δ denote a permutation of $[d]$ such that $|q_{\delta(1)}^\top B p_{\delta(1)}| \geq \dots \geq |q_{\delta(d)}^\top B p_{\delta(d)}|$. Thus, the optimal value of problem (6) is equal to

$$\text{tr}(BX^*) = \sum_{i \in [d]} \sigma_{\delta(i)}(X^*) q_{\delta(i)}^\top B p_{\delta(i)} \leq \sum_{i \in [k]} |q_{\delta(i)}^\top B p_{\delta(i)}| \leq \|B\|_{(k)},$$

where the first inequality is from the fact that $0 \leq \sigma_i(X^*) \leq 1$ for all $i \in [d]$ and $\sum_{i \in [d]} \sigma_i(X^*) \leq k$ and the second one is because $U = [q_{\delta(1)}, \dots, q_{\delta(k)}]$ and $V = [\text{sign}(p_{\delta(1)}^\top B q_{\delta(1)}) p_{\delta(1)}, \dots, \text{sign}(p_{\delta(k)}^\top B q_{\delta(k)}) p_{\delta(k)}]$ is feasible to the maximization problem (1) and yields objective value $\sum_{i \in [k]} |q_{\delta(i)}^\top B p_{\delta(i)}|$.

Thus, problem (6) has optimal value $\|B\|_{(k)}$ with an optimal solution $X^* = \sum_{i \in [k]} \bar{v}_i (\bar{u}_i)^\top$. \square

Introducing binary variables to represent the set variable in SSVD (4) and using Lemma 1 to represent the objective function in SSVD (4), we arrive at the exact MISDP (7) in the following.

Theorem 1 The SSVD (3) can be formulated as

$$\begin{aligned} (\text{SSVD}) \quad w^* := & \max_{z \in \{0,1\}^m, y \in \{0,1\}^n, X \in \mathbb{R}^{n \times m}} \left\{ \text{tr}(AX) : \|X\|_2 \leq 1, \|X\|_* \leq k, \right. \\ & \left. \|X_{:,i}\|_\infty \leq z_i, \forall i \in [m], \sum_{i \in [m]} z_i \leq s_1, \|X_{j,:}\|_\infty \leq y_j, \forall j \in [n], \sum_{j \in [n]} y_j \leq s_2 \right\}. \end{aligned} \quad (7)$$

Proof: See Appendix A.1. \square

We make the following remarks about the result of Theorem 1: (i) since both nuclear norm and induced two-norm are positive semidefinite representable [1], SSVD (7) can be easily recast as an MISDP; (ii) Theorem 1 provides an MISDP formulation for SSVD (3), which can be directly solved by off-the-shelf solvers such as YALMIP, Pajarito, and SCIP-SDP [14]; (iii) based on the MISDP (7), we can design an exact branch-and-cut algorithm for SSVD in Section 6 and in our numerical study, it works well for small- and medium-sized instances (e.g., it can solve the case of $m = n = 500, s_1 = s_2 = 5, k = 2$ within one minute); and (iv) as an important corollary, we derive two equivalent MISDPs for SPCA (5).

4 Approximation Algorithms of SSVD

The results from the previous section are useful to solve small- or medium-sized instances to optimality; however, the exact algorithm might not be scalable to large-scale instances (see our numerical study section for more details). In this section, we develop and analyze three different selection algorithms and two searching algorithms of SSVD (4) with theoretical guarantees for solving the large-scale instances to near optimality.

4.1 Selection Algorithms

We observe that the difficulty of the exact formulations arises from the fact that the representation of Ky Fan k -norm in SSVD (4) is either a nonconvex program or a difficult convex SDP. This motivates us to explore more effective selection criteria to replace the Ky Fan k -norm such that the obtained submatrix by alternative criteria is still a high-quality solution to SSVD (4). Therefore, we design three different selection criteria that are related to the Ky Fan- k norm but are much easier to compute, which correspond to three selection algorithms for SSVD (4). All selection algorithms come with approximation ratios proven to be tight when $s_1 \leq m/2$ and $s_2 \leq n/2$. In particular, the selection algorithms can be implemented into a unified framework, where at each step, using the prespecified criterion, we select a subset of rows or columns to construct a pair of feasible solutions (S_1, S_2) for SSVD (4). However, the approximation ratios of these three selection algorithms require different analyses and thus are derived separately.

Selection Algorithm 1. In the first algorithm, we select a submatrix of size at most $s_1 \times s_2$ that maximizes the Frobenius norm. The detailed implementation can be found in Algorithm 1. We prove a $1/\sqrt{\min\{s_1, s_2\}}$ -approximation ratio of Algorithm 1 in Theorem 2, and this ratio is unimprovable. Note that both the selection procedure and approximation ratio of Algorithm 1 is independent of the parameter k .

Algorithm 1 Selection Algorithm I for SSVD (4)

- 1: **Input:** A matrix $A \in \mathbb{R}^{m \times n}$, integers $s_1 \in [m]$, $s_2 \in [n]$
- 2: Suppose (\hat{S}_1, \hat{S}_2) is an optimal solution to the problem below

$$\max_{S_1 \subseteq [m], S_2 \subseteq [n]} \{\|A_{S_1, S_2}\|_F : |S_1| \leq s_1, |S_2| \leq s_2\}. \quad (8)$$

- 3: **Output:** (\hat{S}_1, \hat{S}_2)
-

Theorem 2 *The selection Algorithm 1 yields a $1/\sqrt{\min\{s_1, s_2\}}$ approximation ratio for SSVD (4), i.e., suppose that the output of Algorithm 1 is (\hat{S}_1, \hat{S}_2) , then*

$$\|A_{\hat{S}_1, \hat{S}_2}\|_{(k)} \geq \frac{1}{\sqrt{\min\{s_1, s_2\}}} w^*,$$

and the ratio is tight (i.e., the best ratio that one can prove) when $s_1 \leq m/2$ and $s_2 \leq n/2$.

Proof: Let us define $\hat{s} := \min\{s_1, s_2\}$, denote an optimal solution of SSVD (4) by (S_1^*, S_2^*) , and define the singular values of the optimal submatrix $A_{S_1^*, S_2^*}$ to be $\sigma_1^* \geq \dots \geq \sigma_{\hat{s}}^* \geq 0$. Correspondingly, for the output (\hat{S}_1, \hat{S}_2) of Algorithm 1, we denote the singular values of the submatrix $A_{\hat{S}_1, \hat{S}_2}$ to be $\hat{\sigma}_1 \geq \dots \geq \hat{\sigma}_{\hat{s}} \geq 0$.

Since (\hat{S}_1, \hat{S}_2) is feasible to SSVD (4), its objective value is equal to

$$\|A_{\hat{S}_1, \hat{S}_2}\|_{(k)} = \sum_{i \in [k]} \hat{\sigma}_i \geq \sqrt{\sum_{i \in [k]} \hat{\sigma}_i^2} \geq \sqrt{\frac{k}{\hat{s}} \sum_{i \in [\hat{s}]} \hat{\sigma}_i^2} = \sqrt{\frac{k}{\hat{s}}} \|A_{\hat{S}_1, \hat{S}_2}\|_F \geq \sqrt{\frac{k}{\hat{s}}} \|A_{S_1^*, S_2^*}\|_F$$

$$= \sqrt{\frac{k}{\hat{s}}} \sqrt{\sum_{i \in [\hat{s}]} (\sigma_i^*)^2} \geq \sqrt{\frac{k}{\hat{s}}} \sqrt{\sum_{i \in [k]} (\sigma_i^*)^2} \geq \frac{1}{\sqrt{\hat{s}}} \sum_{i \in [k]} \sigma_i^* = \frac{1}{\sqrt{\hat{s}}} w^*,$$

where the first and fifth inequalities are because of Cauchy-Schwarz inequality, the second and third equalities are due to the definition of the Frobenius norm, the third inequality is due to the optimality of the solution (\hat{S}_1, \hat{S}_2) to problem (8), and the fourth inequality is due to $k \leq \hat{s}$.

The ratio is tight when $s_1 \leq m/2$ and $s_2 \leq n/2$ since the following example can achieve it.

Example 1 Suppose that $k = s_1 < s_2$, $m = 2s_1$, $n = 2s_2$ and matrix $A \in \mathbb{R}^{m \times n}$ is defined by

$$A = \begin{pmatrix} \mathbf{1}_{s_1, s_2} & \mathbf{0}_{s_1, s_2} \\ \mathbf{0}_{s_1, s_2} & D \end{pmatrix}, D \in \mathbb{R}^{s_1 \times s_2} := \sum_{i \in [k]} \frac{s_1 \sqrt{s_2}}{k} e_i f_i^\top,$$

where $e_i \in \mathbb{R}^{s_1}$ is the i th column of the identity matrix I_{s_1} and $f_i \in \mathbb{R}^{s_2}$ is the i th column of the identity matrix I_{s_2} .

In Example 1, the Algorithm 1 returns a solution $\hat{S}_1 = [s_1]$ and $\hat{S}_2 = [s_2]$ since $\|\mathbf{1}_{s_1, s_2}\|_F = \|D\|_F$, and the objective value of SSVD (4) is $\|\mathbf{1}_{s_1, s_2}\|_{(k)} = \sqrt{s_1 s_2}$. However, the true optimal value is $\|D\|_{(k)} = s_1 \sqrt{s_2}$. This shows that $1/\sqrt{\min\{s_1, s_2\}}$ -approximation ratio is achievable. \square

The selection Algorithm 1 yields the largest approximation ratio (i.e., $1/\sqrt{\min\{s_1, s_2\}}$) for SSVD (4) among all our proposed approximation algorithms; however, it is not polynomially solvable. The computational difficulty of selection Algorithm 1 lies in how to effectively solve problem (8) at Step 2. In fact, problem (8) can be shown to be NP-hard. Fortunately, we prove that the problem (8) can be formulated as a mixed-integer linear program (MILP), making it much more tractable. In addition, problem (8) can be easily solved when A is rectangular diagonal, $s_1 = 1$, or $s_2 = 1$.

Proposition 1 *The problem (8) is NP-hard and can be formulated as the following MILP*

$$\max_{\substack{\mathbf{z} \in [0,1]^{m \times n}, \\ \mathbf{x} \in \{0,1\}^m, \mathbf{y} \in \{0,1\}^n}} \left\{ \sum_{i \in [m]} \sum_{j \in [n]} A_{ij}^2 z_{ij} : z_{ij} \leq x_i, z_{ij} \leq y_j, \forall i \in [m], \forall j \in [n], \sum_{i \in [m]} x_i \leq s_1, \sum_{j \in [n]} y_j \leq s_2 \right\}.$$

Proof: See Appendix A.2. \square

Selection Algorithm 2. The combinatorial optimization problem (8) in Algorithm 1 can be difficult since it jointly selects the two subsets (\hat{S}_1, \hat{S}_2) . To be more scalable, we propose another selection algorithm that selects rows and columns in a separable manner and then combines them to form a submatrix selection for SSVD (4).

Specifically, given a data matrix A , our selection Algorithm 2 runs as follows: (a) first, for each $j \in [n]$, we compute a size- s_1 subset of rows $\hat{S}_1^j \subseteq [m]$ corresponding to the indices of the s_1 largest absolute entries in j th column $A_{:,j}$, and we also let the subset of columns $\hat{S}_2^j \subseteq [n]$ denote the index set of s_2 columns having the largest L_2 norm from submatrix $A_{\hat{S}_1^j, [n]}$, and form a feasible solution $(\hat{S}_1^j, \hat{S}_2^j)$ to SSVD (4); (b) next, we perform the same selection procedure on matrix A by swapping the order of row and column operations and obtain another m feasible solutions to SSVD (4); (c) finally, Algorithm 2 returns the one with the largest Ky Fan k -norm among all the $m+n$ candidate solutions. We derive the approximation ratio of the selection Algorithm 2 for SSVD (4) and further show that this ratio is attainable, as summarized below.

Theorem 3 *The selection Algorithm 2 yields a $1/\sqrt{k \min\{s_1, s_2\}}$ approximation ratio for SSVD (4), i.e., suppose that the output of Algorithm 2 is (\hat{S}_1, \hat{S}_2) , then*

$$\|A_{\hat{S}_1, \hat{S}_2}\|_{(k)} \geq \frac{1}{\sqrt{k \min\{s_1, s_2\}}} w^*,$$

the ratio is tight when $s_1 \leq m/2$ and $s_2 \leq n/2$, and the time complexity of Algorithm 2 is $O((m+n)(m \log(m) + n \log(n) + k s_1 s_2))$.

Algorithm 2 Selection Algorithm for SSVD (4)

```

1: Input: A matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , integers  $s_1 \in [m]$ ,  $s_2 \in [n]$ , and  $1 \leq k \leq \min\{s_1, s_2\}$ 
2: for  $j \in [n]$  do
3:   Let  $\widehat{S}_1^j$  denote the index set of the  $s_1$  largest absolute entries in  $\mathbf{A}_{:,j}$ 
4:    $\widehat{S}_2^j \in \operatorname{argmax}_{S \subseteq [n]} \{\sum_{\ell \in S} \|\mathbf{A}_{\widehat{S}_1^j, \ell}\|_2 : |S| = s_2\}$ 
5: end for
6: for  $i \in [m]$  do
7:   Let  $\widehat{T}_2^i =$  denote the index set of the  $s_2$  largest absolute entries in  $\mathbf{A}_{i,:}$ 
8:    $\widehat{T}_1^i \in \operatorname{argmax}_{T \subseteq [m]} \{\sum_{\ell \in T} \|\mathbf{A}_{\ell, \widehat{T}_2^i}\|_2 : |T| = s_1\}$ 
9: end for
10: Output: The best solution for SSVD (4) among the candidate set  $\{(\widehat{S}_1^j, \widehat{S}_2^j)\}_{j \in [n]} \cup \{(\widehat{T}_1^i, \widehat{T}_2^i)\}_{i \in [m]}$ 

```

Proof: The proof can be divided into three parts: derivation and tightness of the approximation ratio and derivation of the time complexity.

Deriving the Approximation Ratio. We first show a technical result that for any matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$, its largest singular value must satisfy

$$\begin{aligned} \sigma_1(\mathbf{B}) &= \max \left(\sqrt{\lambda_1(\mathbf{B}\mathbf{B}^\top)}, \sqrt{\lambda_1(\mathbf{B}^\top\mathbf{B})} \right) \geq \max \left\{ \max_{i \in [m]} \sqrt{(\mathbf{B}\mathbf{B}^\top)_{ii}}, \max_{j \in [n]} \sqrt{(\mathbf{B}^\top\mathbf{B})_{jj}} \right\} \\ &= \max \left\{ \max_{i \in [m]} \|\mathbf{B}_{i,:}\|_2, \max_{j \in [n]} \|\mathbf{B}_{:,j}\|_2 \right\}, \end{aligned} \quad (9)$$

where the first equality is because $\sigma_1^2(\mathbf{B})$ is the largest eigenvalue of both symmetric matrices $\mathbf{B}\mathbf{B}^\top$ and $\mathbf{B}^\top\mathbf{B}$, the second inequality is because the largest eigenvalue of any symmetric matrix must be no less than its largest diagonal element, and the third equality results from the fact that for matrix $\mathbf{B}\mathbf{B}^\top$ (or $\mathbf{B}^\top\mathbf{B}$), the i th (or j th) diagonal element is equal to the two-norm of the i th row (or j th column) of matrix \mathbf{B} for all $i \in [m]$ (or $j \in [n]$).

Then let us consider the following two combinatorial optimization problems

$$(\bar{S}, j^*) \in \max_{S \subseteq [m], j \in [n]} \{\|\mathbf{A}_{S,j}\|_2 : |S| = s_1\}, (\bar{T}, i^*) \in \max_{S \subseteq [n], i \in [m]} \{\|\mathbf{A}_{i,S}\|_2 : |S| = s_2\}. \quad (10)$$

Next, we will prove the following claim.

Claim For the output $(\widehat{S}_1, \widehat{S}_2)$ of Algorithm 2, we have

$$\|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)} \geq \max\{\|\mathbf{A}_{\bar{S}, j^*}\|_2, \|\mathbf{A}_{i^*, \bar{T}}\|_2\}.$$

Proof: For the j^* -th column of matrix \mathbf{A} , Algorithm 2 selects subsets $(\widehat{S}_1^{j^*}, \widehat{S}_2^{j^*})$ at Steps 3 and 4 and the selected two subsets must satisfy

$$\max_{\ell \in \widehat{S}_2^{j^*}} \|\mathbf{A}_{\widehat{S}_1^{j^*}, \ell}\|_2 = \max_{\ell \in [n]} \|\mathbf{A}_{\widehat{S}_1^{j^*}, \ell}\|_2 \geq \|\mathbf{A}_{\widehat{S}_1^{j^*}, j^*}\|_2 \geq \|\mathbf{A}_{\bar{S}, j^*}\|_2, \quad (11a)$$

where the first equality is due to the optimality of $\widehat{S}_2^{j^*}$ at Step 4, and the second inequality results from the fact that $\widehat{S}_1^{j^*}$ denotes the index set of the s_1 largest absolute entries in $\mathbf{A}_{:,j^*}$.

Similarly, for the corresponding subsets $(\widehat{T}_1^{i^*}, \widehat{T}_2^{i^*})$, we can show that

$$\max_{\ell \in \widehat{T}_1^{i^*}} \|\mathbf{A}_{\ell, \widehat{T}_2^{i^*}}\|_2 \geq \|\mathbf{A}_{i^*, \bar{T}}\|_2. \quad (11b)$$

Since the output $(\widehat{S}_1, \widehat{S}_2)$ of Algorithm 1 is the best solution among the candidates $\{(\widehat{S}_1^j, \widehat{S}_2^j)\}_{j \in [n]} \cup \{(\widehat{T}_1^i, \widehat{T}_2^i)\}_{i \in [m]}$, the output objective value must satisfy

$$\begin{aligned} \|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)} &\geq \max_{S_1, S_2} \left\{ \sigma_1(\mathbf{A}_{S_1, S_2}) : (S_1, S_2) \in \{(\widehat{S}_1^j, \widehat{S}_2^j)\}_{j \in [n]} \cup \{(\widehat{T}_1^i, \widehat{T}_2^i)\}_{i \in [m]} \right\} \\ &\geq \max_{S_1, S_2} \left\{ \max \left\{ \max_{j \in S_2} \|\mathbf{A}_{S_1, j}\|_2, \max_{i \in S_1} \|\mathbf{A}_{i, S_2}\|_2 \right\} : (S_1, S_2) \in \{(\widehat{S}_1^j, \widehat{S}_2^j)\}_{j \in [n]} \cup \{(\widehat{T}_1^i, \widehat{T}_2^i)\}_{i \in [m]} \right\} \\ &\geq \max \left\{ \max_{\ell \in \widehat{S}_2^{j^*}} \|\mathbf{A}_{\widehat{S}_1^{j^*}, \ell}\|_2, \max_{\ell \in \widehat{T}_2^{i^*}} \|\mathbf{A}_{\ell, \widehat{T}_1^{i^*}}\|_2 \right\} \geq \max\{\|\mathbf{A}_{\widehat{S}_1, j^*}\|_2, \|\mathbf{A}_{i^*, \widehat{T}_2}\|_2\}, \end{aligned}$$

where the first inequality is due to the output solution having the largest objective value and $\|\mathbf{A}_{S_1, S_2}\|_{(k)} \geq \sigma_1(\mathbf{A}_{S_1, S_2})$, the second one is due to (9), the third one is from the feasibility of two solutions $(\widehat{S}_1^{j^*}, \widehat{S}_2^{j^*})$ and $(\widehat{T}_1^{i^*}, \widehat{T}_2^{i^*})$, and the last one is from the inequalities (11a) and (11b). \diamond

Now we let (S_1^*, S_2^*) denote an optimal solution to SSVD (4). According to Claim 11, the optimal value is upper bounded by

$$\begin{aligned} w^* = \|\mathbf{A}_{S_1^*, S_2^*}\|_{(k)} &\leq \sqrt{k} \sqrt{\sum_{i \in [k]} \sigma_i^2(\mathbf{A}_{S_1^*, S_2^*})} \leq \sqrt{k} \|\mathbf{A}_{S_1^*, S_2^*}\|_F = \sqrt{k} \sqrt{\sum_{i \in S_1^*} \sum_{j \in S_2^*} A_{ij}^2} \\ &\leq \begin{cases} \sqrt{k s_1} \max_{i \in S_1^*} \|\mathbf{A}_{i, S_2^*}\|_2 \leq \sqrt{k s_1} \|\mathbf{A}_{i^*, \widehat{T}_2}\|_2 \leq \sqrt{k s_1} \|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)} \\ \sqrt{k s_2} \max_{j \in S_2^*} \|\mathbf{A}_{S_1^*, j}\|_2 \leq \sqrt{k s_2} \|\mathbf{A}_{\widehat{S}_1, j^*}\|_2 \leq \sqrt{k s_2} \|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)} \end{cases}, \end{aligned}$$

where the fourth inequality in both lines is from the definitions of (\widehat{T}_2, i^*) and (\widehat{S}_1, j^*) in (10) and the last one on both lines is from Claim 11. Thus, we obtain a $1/\sqrt{k \min\{s_1, s_2\}}$ -approximation ratio.

Tightness of the Approximation Ratio. A worst-case example can be constructed to show that the approximation ratio of Algorithm 2 is tight.

Example 2 For any integers $k > 0, c > 0, t > 0$, suppose that $s_1 = ck, s_2 = 2^t$ such that $\min\{s_1, s_2\} \geq k$, and $m = (2 + s_2)s_1, n = (2 + s_1)s_2$. Then we construct a matrix \mathbf{A} as below

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{0}_{s_1, s_2} & \mathbf{A}_3 \\ \mathbf{0}_{s_1 \times s_1 \times s_2} & \mathbf{0}_{s_1, s_2} & \mathbf{0}_{s_1, s_2} \\ \mathbf{0}_{s_1 \times s_2, s_1 \times s_2} & \mathbf{0}_{s_1 \times s_2, s_2} & \mathbf{A}_2 \end{pmatrix} \in \mathbb{R}^{m \times n},$$

where

$$\mathbf{A}_1 = \begin{pmatrix} \mathbf{1}_{s_2}^\top & \mathbf{0}_{s_2}^\top & \cdots & \mathbf{0}_{s_2}^\top \\ \mathbf{0}_{s_2}^\top & \mathbf{1}_{s_2}^\top & \cdots & \mathbf{0}_{s_2}^\top \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{s_2}^\top & \mathbf{0}_{s_2}^\top & \cdots & \mathbf{1}_{s_2}^\top \end{pmatrix} \in \mathbb{R}^{s_1, s_1 \times s_2}, \mathbf{A}_2 = \begin{pmatrix} \mathbf{1}_{s_1} & \mathbf{0}_{s_1} & \cdots & \mathbf{0}_{s_1} \\ \mathbf{0}_{s_1} & \mathbf{1}_{s_1} & \cdots & \mathbf{0}_{s_1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{s_1} & \mathbf{0}_{s_1} & \cdots & \mathbf{1}_{s_1} \end{pmatrix} \in \mathbb{R}^{s_1 \times s_2, s_2},$$

$$\mathbf{A}_3 = \sum_{i \in [k]} \mathbf{P}_{:,i} \mathbf{H}_{:,i}^\top \in \mathbb{R}^{s_1, s_2}, \mathbf{P} = \begin{pmatrix} \mathbf{1}_{s_1/k} & \mathbf{0}_{s_1/k} & \cdots & \mathbf{0}_{s_1/k} \\ \mathbf{0}_{s_1/k} & \mathbf{1}_{s_1/k} & \cdots & \mathbf{0}_{s_1/k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{s_1/k} & \mathbf{0}_{s_1/k} & \cdots & \mathbf{1}_{s_1/k} \end{pmatrix} \in \mathbb{R}^{s_1, k},$$

and $\mathbf{H} \in \mathbb{R}^{s_2, s_2}$ is a Hadamard matrix of order $s_2 = 2^t$.

Note that in Example 2, the largest absolute value among all the entries in matrix \mathbf{A} is one. Next, we check the selection Algorithm 2 as follows:

- (a) For each $j \in [s_1 \times s_2]$, there is only one entry with value 1 in the j th column of \mathbf{A} and the location of the entry is at $(\lceil j/s_2 \rceil, j)$. Therefore, we can select $\hat{S}_1^j = \{\lceil j/s_2 \rceil\} \cup [s_1 + 2, 2s_1]$; for the row submatrix $\mathbf{A}_{\hat{S}_1^j, [n]}$, the largest two-norm among all of its columns is one. Thus, we can select $\hat{S}_2^j = [(\lceil j/s_2 \rceil - 1) \times s_2 + 1, \lceil j/s_2 \rceil \times s_2]$. The resulting objective value is equal to $\sqrt{s_2}$.
- (b) For each $j \in [s_1 \times s_2 + 1, (s_1 + 1) \times s_2]$, all the entries in the j th column of \mathbf{A} are 0. Thus, we can select $\hat{S}_1^j = [s_1 + 1, 2s_1]$ and $\hat{S}_2^j = [s_1 \times s_2 + 1, s_1 \times s_2 + s_2]$. The resulting objective value is equal to 0.
- (c) For each $j \in [(s_1 + 1) \times s_2 + 1, (s_1 + 2) \times s_2]$, there is a size- s_1 vector with all ones in the column $(\mathbf{A}_2)_{:, j - (s_1 + 1) \times s_2}$. Thus, we can select $\hat{S}_1^j = [2s_1 + (\ell - 1) \times s_1 + 1, 2s_1 + \ell \times s_1]$ where $\ell = j - (s_1 + 1) \times s_2$ and $\hat{S}_2^j = [j - s_2 + 1, j]$. The resulting objective value is equal to $\sqrt{s_1}$.

Similarly, by switching the row and column selections, we can obtain another m feasible solutions to SSVD (4) and the resulting objective value can be $\sqrt{s_1}$, $\sqrt{s_2}$, or 0. Thus, the best objective value output by the selection Algorithm 2 is $\max\{\sqrt{s_1}, \sqrt{s_2}\}$. However, for Example 2, the optimal solution to SSVD (4) is the submatrix \mathbf{A}_3 with optimal value

$$\|\mathbf{A}_3\|_{(k)} = \left\| \sum_{i \in [k]} \mathbf{P}_{:,i} \mathbf{H}_{:,i}^\top \right\|_{(k)} = \sum_{i \in [k]} \|\mathbf{P}_{:,i}\|_2 \|\mathbf{H}_{:,i}\|_2 = k \sqrt{s_1/k} \sqrt{s_2} = \sqrt{k s_1 s_2},$$

where the second equality is due to the fact that $\{\mathbf{P}_{:,i}\}_{i \in [k]}$ and $\{\mathbf{H}_{:,i}\}_{i \in [k]}$ are orthogonal vectors, respectively and Observation 1. Thus, the approximation ratio of Example 2 is

$$\frac{\max\{\sqrt{s_1}, \sqrt{s_2}\}}{\sqrt{k s_1 s_2}} = \frac{1}{\sqrt{k \min\{s_1, s_2\}}},$$

which verifies the tightness of the approximation ratio when $s_1 \leq m/2$ and $s_2 \leq n/2$.

Deriving the Time Complexity. Steps 3 and 4 can be solved by sorting with the time complexity of $m \log(m)$ and $n \log(n)$, respectively. Thus, it takes $O(n(m \log(m) + n \log(n)))$ to implement the first for-loop. Similarly, it takes $O(m(m \log(m) + n \log(n)))$ to implement the second for-loop.

It has been shown in [41] that for any $s_1 \times s_2$ matrix, the computation of its k -truncated SVD can be done in $O(k s_1 s_2)$. Finally, there are $m + n$ solutions to be evaluated. This concludes the complexity analysis. \square

We remark about Theorem 3 that (i) although the approximation ratio of the selection Algorithm 2 is proportional to $1/\sqrt{k}$ and is thus weaker than that of the Algorithm 1 which is independent of k , the Algorithm 2 is of polynomial time, making it much easier-to-implement. This suggests that the Algorithm 2 achieves a better trade-off between algorithm performance and complexity than Algorithm 1, which is further confirmed by our numerical results; and (ii) the selection Algorithm 2 outperforms all the other approximation algorithms in terms of the worst-case performance guarantee for SSVD (4).

Selection Algorithm 3. In this selection algorithm, we trim the top left- and right- singular vectors of matrix \mathbf{A} , as presented in Algorithm 3. Notably, we select the submatrix by specifying a size- s_1 subset of rows that corresponds to the s_1 largest absolute entries of the top left eigenvector, trimming the largest left eigenvector accordingly to be of support size s_1 , and then selecting a size- s_2 subset of columns that corresponds to the largest absolute entries of the product of the trimmed left eigenvector (augmented with zeros if necessary) with the original data matrix \mathbf{A} . We repeat the same procedure for the top right eigenvector. Since there are two candidate feasible solutions, we choose the one with a larger objective value of SSVD (4).

In Theorem 4, we show that selection Algorithm 3 yields a $\sqrt{s_1 s_2}/(k \sqrt{mn})$ -approximation ratio for SSVD (4) with time complexity of $O(m \log(m) + n \log(n) + mn)$. Though Algorithm 3 has the most efficient implementation among the three selection algorithms, its performance is the worst as k increases. This phenomenon is also observed in our numerical study.

Theorem 4 *The selection Algorithm 3 yields a $\sqrt{s_1 s_2}/(k \sqrt{mn})$ approximation ratio for SSVD (4), i.e., suppose that the output of Algorithm 3 is (\hat{S}_1, \hat{S}_2) , then*

$$\|\mathbf{A}_{\hat{S}_1, \hat{S}_2}\|_{(k)} \geq \frac{\sqrt{s_1 s_2}}{k \sqrt{mn}} w^*,$$

Algorithm 3 Selection Algorithm for SSVD (4)

- 1: **Input:** A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, integers $s_1 \in [m]$, $s_2 \in [n]$, and $1 \leq k \leq \min\{s_1, s_2\}$
- 2: Compute left- and right-singular vectors $(\mathbf{u}_1, \mathbf{v}_1)$ corresponding to the largest singular value of \mathbf{A}
- 3: Denote by \hat{S}_1 the indices of s_1 largest absolute entries in \mathbf{u}_1
- 4: Denote by \hat{S}_2 the indices of s_2 largest absolute entries in $(\mathbf{u}_1)_{\hat{S}_1}^\top \mathbf{A}_{\hat{S}_1, [n]}$
- 5: Denote by \hat{T}_2 the indices of s_2 largest absolute entries in \mathbf{v}_1
- 6: Denote by \hat{T}_1 the indices of s_1 largest absolute entries in $\mathbf{A}_{[m], \hat{T}_2} (\mathbf{v}_1)_{\hat{T}_2}$
- 7: **Output:** The better solution of (\hat{S}_1, \hat{S}_2) and (\hat{T}_1, \hat{T}_2)

the ratio is tight when $s_1 \leq m/2$ and $s_2 \leq n/2$, and the time complexity of the selection Algorithm 3 is $O(m \log(m) + n \log(n) + mn)$.

Proof: First, for the output (\hat{S}_1, \hat{S}_2) from Algorithm 3, let us define the vectors $\hat{\mathbf{u}}_1 \in \mathbb{R}^m$ and $\hat{\mathbf{v}}_1 \in \mathbb{R}^n$ as follows

$$(\hat{\mathbf{u}}_1)_i = \begin{cases} (u_1)_i, & \text{if } i \in \hat{S}_1 \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in [m], \hat{\mathbf{u}}_1 = \frac{\hat{\mathbf{u}}_1}{\|\hat{\mathbf{u}}_1\|_2}, (\hat{\mathbf{v}}_1)_j = \begin{cases} (\hat{\mathbf{u}}_1^\top \mathbf{A})_j, & \text{if } j \in \hat{S}_2 \\ 0, & \text{otherwise} \end{cases} \quad \forall j \in [n], \hat{\mathbf{v}}_1 = \frac{\hat{\mathbf{v}}_1}{\|\hat{\mathbf{v}}_1\|_2}.$$

Then we have

$$\sqrt{\frac{s_1}{m}} \sigma_1(\mathbf{A}) = \sqrt{\frac{s_1}{m}} \sigma_1(\mathbf{A}) \|\mathbf{u}_1\|_2 \leq \sigma_1(\mathbf{A}) \hat{\mathbf{u}}_1^\top \mathbf{u}_1 = \hat{\mathbf{u}}_1^\top \mathbf{A} \mathbf{v}_1 \leq \|\hat{\mathbf{u}}_1^\top \mathbf{A}\|_2 \leq \sqrt{\frac{n}{s_2}} \hat{\mathbf{u}}_1^\top \mathbf{A} \hat{\mathbf{v}}_1,$$

where the first inequality is because $\hat{\mathbf{u}}_1^\top \mathbf{u}_1 = \sqrt{\sum_{i \in \hat{S}_1} (u_1)_i^2} \geq \sqrt{s_1/m} \sqrt{\sum_{i \in [m]} (u_1)_i^2}$, the second equality is from $\mathbf{A} \mathbf{v}_1 = \sigma_1(\mathbf{A}) \mathbf{u}_1$, the second inequality is from Cauchy-Schwarz inequality and $\|\mathbf{v}_1\|_2 = 1$, and the last one follows the similar reasoning as the first inequality.

Now we have

$$w^* \leq \|\mathbf{A}\|_{(k)} \leq k \sigma_1(\mathbf{A}) \leq k \frac{\sqrt{mn}}{\sqrt{s_1 s_2}} \hat{\mathbf{u}}_1^\top \mathbf{A} \hat{\mathbf{v}}_1 = k \frac{\sqrt{mn}}{\sqrt{s_1 s_2}} (\hat{\mathbf{u}}_1)_{\hat{S}_1}^\top \mathbf{A}_{\hat{S}_1, \hat{S}_2} (\hat{\mathbf{v}}_1)_{\hat{S}_2} \leq k \frac{\sqrt{mn}}{\sqrt{s_1 s_2}} \sigma_1(\mathbf{A}_{\hat{S}_1, \hat{S}_2}),$$

where the first equality is because $(\hat{\mathbf{u}}_1)_i = 0$ for all $i \in [m] \setminus \hat{S}_1$ and $(\hat{\mathbf{v}}_1)_j = 0$ for all $j \in [n] \setminus \hat{S}_2$ and the last inequality is due to Part (i) in Lemma 1 with matrix $\mathbf{A}_{\hat{S}_1, \hat{S}_2}$ and $k = 1$.

Similarly, for the output (\hat{T}_1, \hat{T}_2) from Algorithm 3, we also have

$$w^* \leq k \frac{\sqrt{mn}}{\sqrt{s_1 s_2}} \sigma_1(\mathbf{A}_{\hat{T}_1, \hat{T}_2}),$$

This concludes that the approximation ratio of selection Algorithm 3 is equal to $\sqrt{s_1 s_2} / (k \sqrt{mn})$.

For the following example, the output of the selection Algorithm 3 achieves the ratio of $\sqrt{s_1 s_2} / (k \sqrt{mn})$.

Example 3 For any positive integers $k > 0, t_1 > 0, t_2 > 0$, suppose that $s_1 = 2^{t_1}$, $s_2 = 2^{t_2}$ such that $\min\{s_1, s_2\} \geq k + 1$, and $m = 2s_1, n = 2s_2$, then the matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ is constructed by

$$\mathbf{A} = \sum_{i \in [k]} \frac{\sqrt{mn}}{\sqrt{s_1 s_2}} \begin{pmatrix} \mathbf{0}_{s_1} \\ \mathbf{H}_{:,i} \end{pmatrix} \begin{pmatrix} \mathbf{0}_{s_2} \\ \mathbf{F}_{:,i} \end{pmatrix}^\top + \begin{pmatrix} \mathbf{1}_{s_1} \\ \mathbf{H}_{:,k+1} \end{pmatrix} \begin{pmatrix} \mathbf{1}_{s_2} \\ \mathbf{F}_{:,k+1} \end{pmatrix}^\top,$$

where $\mathbf{H} \in \mathbb{R}^{s_1, s_1}$ and $\mathbf{F} \in \mathbb{R}^{s_2, s_2}$ are Hadamard matrices of order 2^{t_1} and order 2^{t_2} , respectively.

In Example 3, by the construction and the fact that the columns of a Hadamard matrix are orthogonal, the $(k+1)$ left- and right- vectors defining matrix \mathbf{A} are orthogonal, respectively. It follows that all the non-zero singular values of \mathbf{A} are equal to \sqrt{mn} as $m = 2s_1, n = 2s_2$, and a pair of top singular vectors can be

$$\mathbf{u}_1 = \frac{1}{\sqrt{m}} \begin{pmatrix} \mathbf{1}_{s_1} \\ \mathbf{H}_{:,k+1} \end{pmatrix}, \mathbf{v}_1 = \frac{1}{\sqrt{n}} \begin{pmatrix} \mathbf{1}_{s_2} \\ \mathbf{F}_{:,k+1} \end{pmatrix}.$$

Given vectors \mathbf{u}_1 and \mathbf{v}_1 , Algorithm 3 selects $\hat{S}_1 = [s_1]$ at Step 3 and $\hat{T}_2 = [s_2]$ at Step 5, respectively.

Then the submatrix $\mathbf{A}_{\hat{S}_1, [n]}$ and submatrix $\mathbf{A}_{[m], \hat{T}_2}$ can be written as

$$\mathbf{A}_{\hat{S}_1, [n]} = (\mathbf{1}_{s_1, s_2} \mathbf{1}_{s_1} \mathbf{F}_{:,k+1}^\top), \mathbf{A}_{[m], \hat{T}_2} = \begin{pmatrix} \mathbf{1}_{s_1, s_2} \\ \mathbf{H}_{:,k+1} \mathbf{1}_{s_2}^\top \end{pmatrix}.$$

In this way, the s_2 largest absolute entries in $(\mathbf{u}_1)_{\hat{S}_1}^\top \mathbf{A}_{\hat{S}_1, [n]}$ can be achieved by the s_2 first entries so $\hat{S}_2 = [s_2]$. Similarly, Algorithm 3 chooses $\hat{T}_1 = [s_1]$ at Step 6.

For the two solutions (\hat{S}_1, \hat{S}_2) and (\hat{T}_1, \hat{T}_2) , both objective values of SSVD (4) are equal to $\|\mathbf{A}_{[s_1], [s_2]}\|_{(k)} = \|\mathbf{1}_{s_1, s_2}\|_{(k)} = \sqrt{s_1 s_2}$.

However, the true optimal submatrix of SSVD (4) is $\mathbf{A}_{[s_1+1, 2s_1], [s_2+1, 2s_2]}$ with the optimal value

$$\|\mathbf{A}_{[s_1+1, 2s_1], [s_2+1, 2s_2]}\|_{(k)} = \left\| \sum_{i \in [k]} \frac{\sqrt{mn}}{\sqrt{s_1 s_2}} \mathbf{H}_{:,i} \mathbf{F}_{:,i}^\top + \mathbf{H}_{:,k+1} \mathbf{F}_{:,k+1}^\top \right\|_{(k)} = k\sqrt{mn}.$$

Thus, for Example 3, the approximation ratio of selection Algorithm 3 exactly is $\sqrt{s_1 s_2} / (k\sqrt{mn})$.

Finally, the top singular vectors can be obtained using 1-truncated SVD of matrix \mathbf{A} , which takes $O(mn)$ according to [41]. Next, Steps 3 and 4 can be solved by sorting with the time complexity of $m \log(m)$ and $n \log(n)$, respectively. So do Steps 5 and 6. \square

It is worthy of mentioning that: (i) when $k = 1$, the approximation ratios of selection Algorithm 1, Algorithm 2, and Algorithm 3 match the best-known approximation ratio of rank-one SPCA and rank-one SSVD (see, e.g., [4, 23]); (ii) the approximation ratios of the three selection algorithms satisfy Algorithm 1 \geq Algorithm 2 \geq Algorithm 3, while their scalabilities are in the reverse order. This indicates a clear trade-off between optimality and time complexity when employing them to solve practical problems; and (iii) we see in Theorem 4 that the approximation ratio of Algorithm 3 is increasing in $s_1 s_2$ and decreasing in mn . Thus, when $s_1 \leq m/2$ and $s_2 \leq n/2$ holds, the selection Algorithm 3 yields the worst approximation ratio among our approximation algorithms. On the other hand, when s_1, s_2 are very large, the selection Algorithm 3 can complement and dominate the other algorithms. For example, if $s_1 = m, s_2 = n$, and $k = 1$, the approximation ratio of Algorithm 3 attains one, superior to the other ones. Besides, the selection Algorithm 3 is the most efficient according to our numerical study, taking one second on average.

4.2 Searching Algorithms

This subsection studies the widely-used greedy and local search algorithms (see, e.g., [12, 24]). We will show that the two searching algorithms have the same approximation ratio for solving SSVD (4), and the ratio is tight for the cases of $s_1 \leq m/2$ and $s_2 \leq n/2$. Notably, both searching algorithms aim to improve the current solution through different searching strategies to explore new solutions of SSVD (4). Thus, we can unify the two search algorithms, where at each step, based on the current solution and using the pre-specified searching strategy, the algorithm searches for a new solution to replace the current one if such a movement improves the objective value of SSVD (4). Next, we provide implementation details and theoretical guarantees for the greedy and local searching algorithms, respectively.

Greedy Algorithm 4. The greedy algorithm for SSVD (4) proceeds as follows: At each iteration, given two subsets denoting the selected rows and columns, respectively, the algorithm first searches for a new row from the unchosen set to maximize the objective value of the augmented submatrix and adds the optimal

row to the row subset, then finds a new column to update the column subset until the two subsets attaining sizes s_1 and s_2 , respectively. The implementation details can be found in Algorithm 4.

Theorem 5 shows a $1/\sqrt{ks_1s_2}$ -approximation ratio and a $O(\max\{s_1, s_2\}(m+n)ks_1s_2)$ -time complexity when employing the greedy Algorithm 4 to solve SSVD (4).

Algorithm 4 Greedy Algorithm for SSVD (4)

```

1: Input: A matrix  $A \in \mathbb{R}^{m \times n}$ , integers  $s_1 \in [m]$ ,  $s_2 \in [n]$ , and  $1 \leq k \leq \min\{s_1, s_2\}$ 
2: Compute  $(i^*, j^*) \in \operatorname{argmax}_{i \in [m], j \in [n]} \{|A_{i,j}|\}$ 
3: Define subsets  $\hat{S}_1 := \{i^*\}$  and  $\hat{S}_2 := \{j^*\}$ 
4: for  $\ell = 2, \dots, \max\{s_1, s_2\}$  do
5:   if  $\ell \leq \min\{s_1, s_2\}$  then
6:      $i^* \in \operatorname{argmax}_{i \in [m] \setminus \hat{S}_1} \|A_{\hat{S}_1 \cup \{i\}, \hat{S}_2}\|_{(\min\{\ell, k\})}$ 
7:     Update  $\hat{S}_1 := \hat{S}_1 \cup \{i^*\}$ 
8:      $j^* \in \operatorname{argmax}_{j \in [n] \setminus \hat{S}_2} \|A_{\hat{S}_1, \hat{S}_2 \cup \{j\}}\|_{(\min\{\ell, k\})}$ 
9:     Update  $\hat{S}_2 := \hat{S}_2 \cup \{j^*\}$ 
10:   else if  $s_1 \leq s_2$  then
11:      $j^* \in \operatorname{argmax}_{j \in [n] \setminus \hat{S}_2} \|A_{\hat{S}_1, \hat{S}_2 \cup \{j\}}\|_{(k)}$ 
12:     Update  $\hat{S}_2 := \hat{S}_2 \cup \{j^*\}$ 
13:   else
14:      $i^* \in \operatorname{argmax}_{i \in [m] \setminus \hat{S}_1} \|A_{\hat{S}_1 \cup \{i\}, \hat{S}_2}\|_{(k)}$ 
15:     Update  $\hat{S}_1 := \hat{S}_1 \cup \{i^*\}$ 
16:   end if
17: end for
18: Output:  $\hat{S}_1, \hat{S}_2$ 

```

Theorem 5 The greedy Algorithm 4 yields a $1/\sqrt{ks_1s_2}$ -approximation ratio for SSVD (4), i.e., suppose that the output of Algorithm 4 is (\hat{S}_1, \hat{S}_2) , then

$$\|A_{\hat{S}_1, \hat{S}_2}\|_{(k)} \geq \frac{1}{\sqrt{ks_1s_2}} w^*,$$

the ratio is tight when $s_1 \leq m/2$ and $s_2 \leq n/2$, and the time complexity of the greedy Algorithm 4 is $O(\max\{s_1, s_2\}(m+n)ks_1s_2)$.

Proof: Suppose that (S_1^*, S_2^*) is an optimal solution to SSVD (4). According to the last two equalities and inequalities in the proof of Theorem 2, we have

$$w^* = \|A_{S_1^*, S_2^*}\|_{(k)} \leq \sqrt{k} \|A_{S_1^*, S_2^*}\|_F = \sqrt{k} \sqrt{\sum_{i \in S_1^*} \sum_{j \in S_2^*} A_{ij}^2} \leq \sqrt{k} \sqrt{s_1 s_2} \max_{i \in [m], j \in [n]} |A_{ij}|,$$

where the first inequality can be found in the proof of Theorem 2, the right-hand side is upper bounded by the output value from greedy Algorithm 4, and thus we obtain the $1/\sqrt{ks_1s_2}$ -approximation ratio. The proof of its tightness is based on the example below.

Example 4 For any positive integers $k > 0, t > 0$, suppose that $s_1 = ck, s_2 = 2^t$ such that $\min\{s_1, s_2\} \geq k$, and $m = 2s_1, n = 2s_2$. Let us construct matrix $A \in \mathbb{R}^{m \times n}$ as

$$A = \begin{pmatrix} 1 & \mathbf{0}_{s_2-1}^\top & \mathbf{0}_{s_2}^\top \\ \mathbf{0}_{s_1-1} & \mathbf{0}_{s_1-1, s_2-1} & \mathbf{0}_{s_1-1, s_2} \\ \mathbf{0}_{s_1} & \mathbf{0}_{s_1, s_2-1} & A_3 \in \mathbb{R}^{s_1, s_2} \end{pmatrix},$$

where matrix \mathbf{A}_3 is defined in Example 2.

For Example 4, the greedy Algorithm 4 starts with $\hat{S}_1 = \{1\}$ and $\hat{S}_2 = \{1\}$ since 1 is the largest absolute entry in matrix \mathbf{A} . Then at each iteration ℓ , we update $\hat{S}_1 := \hat{S}_1 \cup \{i^*\}$ or $\hat{S}_2 := \hat{S}_2 \cup \{j^*\}$ or both depending on the values of s_1 and s_2 . Hence, in Example 4, the greedy Algorithm 4 outputs the objective value of SSVD (4) as

$$\|\mathbf{A}_{\hat{S}_1, \hat{S}_2}\|_{(k)} = \|\mathbf{A}_{[s_1], [s_2]}\|_{(k)} = 1.$$

On the other hand, the true optimal value of SSVD (4) is $\|\mathbf{A}_3\|_{(k)} = \sqrt{ks_1s_2}$. This proves the tightness when $s_1 \leq m/2$ and $s_2 \leq n/2$.

As for the time complexity, it is known that at most $\max\{s_1, s_2\}$ iterations are required in the for-loop of the greedy Algorithm 4. At each iteration, solving the two maximization problems is equivalent to computing k -truncated SVD of a submatrix of size no larger than $s_1 \times s_2$ at most $(m+n)$ times. According to [41], the time complexity of solving k -truncated SVD of a size $s_1 \times s_2$ matrix is $O(ks_1s_2)$. Hence, each iteration takes $O((m+n)ks_1s_2)$ operations. This completes the proof of time complexity. \square

According to Theorem 5, the value ks_1s_2 plays an important role in both theoretical ratio and computational complexity of the greedy Algorithm 4, similar to the selection Algorithm 2. Our numerical study also confirms this phenomenon. That is, the solutions returned by Algorithm 4 and Algorithm 2 are often non-comparable, and their running times are quite close. Besides, the greedy Algorithm 4 is often more efficient than the local search Algorithm 5, although both algorithms achieve the same worst-case guarantees. Nevertheless, our numerical study shows that the local search Algorithm 5 consistently delivers better solutions.

Local Search Algorithm 5. The local search algorithm, one of the well-known improved heuristic methods, aims to find a better solution by exploring neighborhoods. Specifically, the local search algorithm for SSVD (4) proceeds as follows: (i) initialize a feasible solution (\hat{S}_1, \hat{S}_2) with the output of the greedy Algorithm 4, (ii) at each iteration, first swap an element in \hat{S}_1 with an element in $[m] \setminus \hat{S}_1$ and check if such a movement increases the objective value or not- if YES, then accept the swap, and otherwise continue, and (iii) explore the neighborhood solutions by swapping elements between \hat{S}_2 and $[n] \setminus \hat{S}_2$. By doing so, the two subsets \hat{S}_1, \hat{S}_2 will be updated separately. More details are presented in Algorithm 5. Surprisingly, Theorem 6 shows that the local search Algorithm 5 also yields a tight $1/\sqrt{ks_1s_2}$ -approximation ratio. That is, in the worst-case, the local search Algorithm 5 might not be able to improve the performance of the greedy Algorithm 4 when $s_1 \leq m/2$ and $s_2 \leq n/2$, as illustrated in our numerical study.

Theorem 6 *The local search Algorithm 5 yields a $1/\sqrt{ks_1s_2}$ -approximation ratio for SSVD (4), i.e., suppose that the output of Algorithm 5 is (\hat{S}_1, \hat{S}_2) , then*

$$\|\mathbf{A}_{\hat{S}_1, \hat{S}_2}\|_{(k)} \geq \frac{1}{\sqrt{ks_1s_2}} w^*,$$

and the ratio is tight when $s_1 \leq m/2$ and $s_2 \leq n/2$.

Proof: Since local search Algorithm 5 uses the output solution of greedy Algorithm 4 as the initial solution and local search Algorithm 5 is an improved heuristic method, the approximation ratio of Algorithm 5 must be no worse than that of greedy Algorithm 4, i.e., $1/\sqrt{ks_1s_2}$.

Moreover, we show that the $1/\sqrt{ks_1s_2}$ -approximation ratio is also tight for local search Algorithm 5 and the ratio can be achieved by Example 4. Indeed, in Example 4, swapping one element from the chosen set and one element from the unchosen set, the resulting objective value is still equal to 1 and does not improve the current solution. It follows that the output of local search Algorithm 5 is the same as that of Algorithm 4 with the objective value 1, which yields $1/\sqrt{ks_1s_2}$ -approximation ratio for SSVD (4). \square

Algorithm 5 Local Search Algorithm for SSVD (4)

```

1: Input: A matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , integers  $s_1 \in [m]$ ,  $s_2 \in [n]$ , and  $1 \leq k \leq \min\{s_1, s_2\}$ 
2: Initialize  $(\widehat{S}_1, \widehat{S}_2)$  as the output of greedy Algorithm 4
3: do
4:   for each pair  $(i_1, j_1) \in \widehat{S}_1 \times ([m] \setminus \widehat{S}_1)$  do
5:     if  $\|\mathbf{A}_{\widehat{S}_1 \cup \{j_1\} \setminus \{i_1\}, \widehat{S}_2}\|_{(k)} > \|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)}$  then
6:       Update  $\widehat{S}_1 := \widehat{S}_1 \cup \{j_1\} \setminus \{i_1\}$ 
7:     end if
8:   end for
9:   for each pair  $(i_2, j_2) \in \widehat{S}_2 \times ([n] \setminus \widehat{S}_2)$  do
10:    if  $\|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2 \cup \{j_2\} \setminus \{i_2\}}\|_{(k)} > \|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)}$  then
11:      Update  $\widehat{S}_2 := \widehat{S}_2 \cup \{j_2\} \setminus \{i_2\}$ 
12:    end if
13:  end for
14: while there is still an improvement
15: Output:  $\widehat{S}_1, \widehat{S}_2$ 

```

It is interesting to note that although the theoretical approximation ratios of the greedy Algorithm 4 and local search Algorithm 5 seem not as good as selection Algorithm 1 and Algorithm 2, our numerical study shows that the searching algorithms, in general, work better and can find a near-optimal solution for most of the testing cases, especially that the Algorithm 5 is very effective in solving the large-scale instances.

For the Steps 5 and 10 of the local search Algorithm 5, we multiply the right-hand side of the strict inequalities by a factor $(1 + \delta)$, where $\delta > 0$ denotes a strict improvement factor. Then the local search Algorithm 5 possesses polynomial time complexity. Note that since the local search Algorithm 5 admits the same approximation ratio as the greedy Algorithm 4, the extra factor $(1 + \delta)$ does not affect the approximation ratio of the local search Algorithm 5.

Proposition 2 *Time complexity of the revised local search Algorithm 5 is $O((nks_1^2s_2 + mks_1s_2^2)L/\delta)$ where $\delta > 0$ denotes the strict improvement factor and $L := \lceil \log_2 \sqrt{ks_1s_2} \rceil + \ell_1 + \ell_2$ with ℓ_1 denoting the maximum binary encoding length of the absolute value of each entry in matrix \mathbf{A} and ℓ_2 denoting the encoding length of the smallest fraction.*

Proof: First, according to the proof of Theorem 5, we can derive that for any submatrix $\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}$, the following always holds

$$\|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)} \leq \sqrt{ks_1s_2} \max_{i \in [m], j \in [n]} |\mathbf{A}_{ij}|.$$

As ℓ_1 is the maximum binary encoding length of $|\mathbf{A}_{ij}|$ for each $i \in [m], j \in [n]$, $\|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)}$ can be further upper bounded by

$$\|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)} \leq 2^{\ell_1} \sqrt{ks_1s_2}.$$

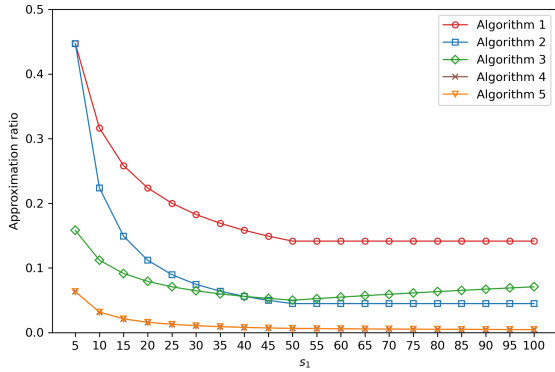
According to the definition of ℓ_2 , the largest and smallest value of $\|\mathbf{A}_{\widehat{S}_1, \widehat{S}_2}\|_{(k)}$ can be $2^{\ell_1} \sqrt{ks_1s_2}$ and $2^{-\ell_2}$. Suppose that we accept the swapping if the improvement ratio is at least $1 + \delta$, the number of swapping steps of local search Algorithm 5 is at most

$$\log_{1+\delta}(2^{\ell_1+\ell_2} \sqrt{ks_1s_2}) \leq \frac{2}{\delta} L,$$

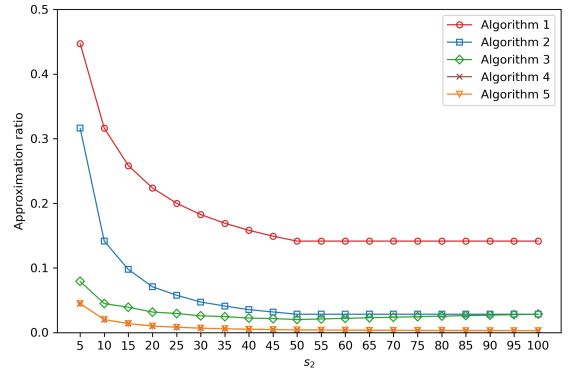
where $L := \lceil \log_2 \sqrt{ks_1s_2} \rceil + \ell_1 + \ell_2$.

Finally, for each swapping at Steps 4-13, there are at most $(ns_1 + ms_2)$ distinct evaluations of the k -truncated SVD of a size $s_1 \times s_2$ submatrix, which takes at most $O(nks_1^2s_2 + mks_1s_2^2)$. Therefore, local search Algorithm 5 has an overall time complexity $O((nks_1^2s_2 + mks_1s_2^2)L/\delta)$. \square

Figure 1 illustrates the approximation ratios in Theorems 2-6, where we fix (m, n, s_2) and set k to be the floor of $\min\{s_1, s_2\}/5$ in Figure 1a, and we fix (m, n, s_1) and set k to be the floor of $\min\{s_1, s_2\}/2$ in Figure 1b. It is seen from Figures 1a and 1b that (i) the selection Algorithm 1 has the largest ratio which is independent of k ; (ii) the approximation ratio of selection Algorithm 2 is sensitive to (s_1, s_2, k) , especially to the k , which is evident in the point-to-point decline of the blue rectangle line from Figure 1a to Figure 1b with different k ; (iii) as previously stated, the selection Algorithm 3 is the only one that can be increasing in s_1s_2 . Hence, the ratio in green diamond line can match or be larger than that of Algorithm 2 when s_1 or s_2 is increased to m or n in Figure 1; and (iv) the searching algorithms return the smallest approximation ratios for the parameters considered in Figure 1.



(a) $(m, n, s_2) = (100, 100, 50)$ and $k = \lfloor \min\{s_1, s_2\}/5 \rfloor$



(b) $(m, n, s_1) = (100, 100, 50)$ and $k = \lfloor \min\{s_1, s_2\}/2 \rfloor$

Fig. 1: Approximation ratios of algorithm 1-5 given different parameters (m, n, s_1, s_2, k)

5 Extensions to Row-Sparse PCA

The general rank- k SPCA (5) is closely related to SSVD (4), which has been recently studied in [12, 45]. The key ingredient of SPCA (5) is that its data matrix $\mathbf{A} \in \mathbb{S}_+^n$ is symmetric and positive semidefinite. We show that SPCA (5) can be viewed as a special case of SSVD (3). It should be noted that matrix \mathbf{A} is assumed to be positive semidefinite for SPCA but can be non-symmetric for SSVD (3).

Proposition 3 *When matrix \mathbf{A} is positive semidefinite and $s_1 = s_2 = s$, SSVD (3) reduces to SPCA (5), i.e., there exists an optimal solution $(\mathbf{U}^*, \mathbf{V}^*)$ to SSVD (3) satisfying $\mathbf{U}^* = \mathbf{V}^*$.*

Proof: When $\mathbf{A} \in \mathbb{S}_+^n$ and $s_1 = s_2 = s$, SSVD (3) becomes

$$\max_{\mathbf{U} \in \mathbb{R}^{n \times k}, \mathbf{V} \in \mathbb{R}^{n \times k}} \{ \text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{V}) : \mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}_k, \|\mathbf{U}\|_0 \leq s, \|\mathbf{V}\|_0 \leq s \}. \quad (12)$$

Then for any feasible solution (\mathbf{U}, \mathbf{V}) to problem (12), suppose that for each $i \in [k]$, \mathbf{u}_i and \mathbf{v}_i are the i th columns of \mathbf{U} and \mathbf{V} , respectively, then the objective value satisfies

$$\text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{V}) = \sum_{i \in [k]} \mathbf{u}_i^\top \mathbf{A} \mathbf{v}_i = \sum_{i \in [k]} \mathbf{u}_i^\top \mathbf{A}^{1/2} \mathbf{A}^{1/2} \mathbf{v}_i \leq \sum_{i \in [k]} \sqrt{\mathbf{u}_i^\top \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} \mathbf{v}_i}$$

$$\leq \sum_{i \in [k]} \frac{1}{2} (\mathbf{u}_i^\top \mathbf{A} \mathbf{u}_i + \mathbf{v}_i^\top \mathbf{A} \mathbf{v}_i) = \frac{1}{2} (\text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{U}) + \text{tr}(\mathbf{V}^\top \mathbf{A} \mathbf{V})) \leq \max \{ \text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{U}), \text{tr}(\mathbf{V}^\top \mathbf{A} \mathbf{V}) \}.$$

The above analysis states that for any feasible solution (\mathbf{U}, \mathbf{V}) to problem (12), there exists a better solution between (\mathbf{U}, \mathbf{U}) and (\mathbf{V}, \mathbf{V}) . This implies that there must exist $\mathbf{U}^* = \mathbf{V}^*$ at optimality. \square

Using Proposition 3 and the fact that the rank-one SPCA has been proven to be NP-hard and inapproximable in [25], we prove the following complexity result for SSVD (3).

Corollary 1 *The SSVD (3) is NP-hard to solve and is inapproximable by any polynomial-time algorithm with a constant-factor approximation ratio.*

Following SSVD (4), we also derive an equivalent combinatorial formulation for SPCA (5), i.e.,

$$(\text{SPCA}) \quad w^{spca} := \max_{S \subseteq [n], |S| \leq s} \|\mathbf{A}_{S,S}\|_{(k)}. \quad (13)$$

5.1 Exact Formulations of SPCA

Following the spirit of Theorem 1, we can also derive two exact MISDP formulations for SPCA (5). Since Theorem 1 replies on Lemma 1 that reformulates the Ky Fan k -norm of a matrix as an SDP, we next show the similar result in Lemma 2 for the positive semidefinite matrix.

Lemma 2 *For any matrix $\mathbf{B} \in \mathcal{S}_+^n$ with rank r and an integer $k \in [r]$, let $\mathbf{B} = \sum_{i \in [r]} \lambda_i(\mathbf{B}) \mathbf{q}_i \mathbf{q}_i^\top$ denote the eigen-decomposition of matrix \mathbf{B} . Then we have*

$$\max_{\mathbf{X} \in \mathcal{S}_+^n} \{ \text{tr}(\mathbf{B} \mathbf{X}) : \mathbf{X} \preceq \mathbf{I}_n, \text{tr}(\mathbf{X}) \leq k \} = \|\mathbf{B}\|_{(k)},$$

where the equality can be achieved by $\mathbf{X}^* = \sum_{i \in [k]} \mathbf{q}_i \mathbf{q}_i^\top$.

Proof: According to Lemma 1, $\|\mathbf{B}\|_{(k)}$ can be formulated by

$$\max_{\mathbf{X} \in \mathbb{R}^{n \times n}} \{ \text{tr}(\mathbf{B} \mathbf{X}) : \|\mathbf{X}\|_2 \leq 1, \|\mathbf{X}\|_* \leq k \}, \quad (14)$$

which relaxes the maximization problem in Lemma 2.

Given that the optimal value of problem (14) is equal to $\|\mathbf{B}\|_{(k)}$, the optimal value of the maximization problem in Lemma 2 must be upper bounded by $\|\mathbf{B}\|_{(k)}$. On the other hand, the upper bound $\|\mathbf{B}\|_{(k)}$ is achievable by the feasible solution $\mathbf{X}^* = \sum_{i \in [k]} \mathbf{q}_i \mathbf{q}_i^\top$ since

$$\text{tr}(\mathbf{B} \mathbf{X}^*) = \sum_{i \in [k]} \mathbf{q}_i^\top \mathbf{B} \mathbf{q}_i = \sum_{i \in [k]} \lambda_i(\mathbf{B}) = \|\mathbf{B}\|_{(k)},$$

where the last equality is because for any positive semidefinite matrix, its singular values are equal to the eigenvalues. This completes the proof. \square

We are ready to derive the first exact MISDP for SPCA (5).

Corollary 2 *The SPCA (5) is equivalent to*

$$(\text{SPCA}) \quad w^{spca} := \max_{\mathbf{z} \in \{0,1\}^n, \mathbf{X} \in \mathcal{S}_+^n} \left\{ \text{tr}(\mathbf{A} \mathbf{X}) : \mathbf{X} \preceq \mathbf{I}_n, \text{tr}(\mathbf{X}) \leq k, X_{ii} \leq z_i, \forall i \in [n], \sum_{i \in [n]} z_i \leq s \right\}. \quad (15)$$

Proof: The result directly follows from the proof of Theorem 1 and the identity in Lemma 2. Thus, the proof is omitted for brevity. \square

For matrix $\mathbf{A} \in \mathcal{S}_+^n$ with rank r , let $\mathbf{A} = \mathbf{C}^\top \mathbf{C}$ denote the Cholesky decomposition, where $\mathbf{C} \in \mathbb{R}^{r \times n}$ is the corresponding Cholesky matrix and for each $i \in [n]$ with $\mathbf{c}_i \in \mathbb{R}^r$ denoting its i th column. Then, we obtain another exact MISDP for SPCA (5).

Corollary 3 *The SPCA (5) is equivalent to*

$$\begin{aligned} (\text{SPCA}) \quad w^{spca} := & \max_{\substack{\mathbf{z} \in \{0,1\}^n, \mathbf{X} \in \mathcal{S}_+^r, \\ \mathbf{W}_1, \dots, \mathbf{W}_n \in \mathcal{S}_+^r}} \left\{ \sum_{i \in [n]} \mathbf{c}_i^\top \mathbf{W}_i \mathbf{c}_i : \mathbf{X} \preceq \mathbf{I}_r, \text{tr}(\mathbf{X}) \leq k, \mathbf{X} \succeq \mathbf{W}_i, \forall i \in [n], \right. \\ & \left. \text{tr}(\mathbf{W}_i) \leq z_i, \forall i \in [n], \sum_{i \in [n]} z_i \leq s \right\}. \end{aligned} \quad (16)$$

Proof: Given $\mathbf{A} = \mathbf{C}^\top \mathbf{C}$, for any feasible subset S to SPCA (13), its objective value is equal to

$$\|\mathbf{A}_{S,S}\|_{(k)} = \left\| \mathbf{C}_{[r],S}^\top \mathbf{C}_{[r],S} \right\|_{(k)} = \left\| \mathbf{C}_{[r],S} \mathbf{C}_{[r],S}^\top \right\|_{(k)} = \left\| \sum_{i \in S} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)},$$

where the second equality is because for any matrix \mathbf{B} , $\mathbf{B}^\top \mathbf{B}$ and $\mathbf{B}\mathbf{B}^\top$ are positive semidefinite and have the same non-zero eigenvalues.

Therefore, SPCA (13) can be formulated as

$$\max_{S \subseteq [n], |S| \leq s} \left\| \sum_{i \in S} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)}.$$

Introducing binary variables $z \in \mathbb{R}^n$ such that for all $i \in [n]$, let $z_i = 1$ if $i \in S$ and zero; otherwise, the above formulation can be written as

$$\max_{z \in \mathbb{R}^n} \left\{ \left\| \sum_{i \in [n]} z_i \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)} : \sum_{i \in [n]} z_i \leq s \right\}.$$

Using the result in Lemma 2 and plugging the SDP formulation of the objective function above, the SPCA now is equivalent to

$$\max_{z \in \mathbb{R}^n, \mathbf{X} \in \mathcal{S}_+^r} \left\{ \text{tr} \left(\sum_{i \in [n]} z_i \mathbf{X} \mathbf{c}_i \mathbf{c}_i^\top \right) : \mathbf{X} \preceq \mathbf{I}_r, \text{tr}(\mathbf{X}) \leq k, \sum_{i \in [n]} z_i \leq s \right\}.$$

Following the proof of theorem 1 in [23], the bilinear terms $\{z_i \mathbf{X}\}_{i \in [n]}$ can be linearized and finally, the SPCA (5) can be formulated by MISDP (16). \square

5.2 Selection Algorithms of SPCA

As a special case of SSVD (4), it is natural to extend the proposed approximation algorithms of SSVD (4) to solve SPCA (13). In this subsection, we tailor the selection algorithms to solve SPCA (13). In particular, the extensions of selection algorithms are built on the following key lemma.

Lemma 3 *For any matrix $\mathbf{B} \in \mathcal{S}_+^n$ and two equal-sized subsets $S_1, S_2 \subseteq [n]$, we have*

$$\|\mathbf{B}_{S_1, S_2}\|_{(k)} \leq \max\{\|\mathbf{B}_{S_1, S_1}\|_{(k)}, \|\mathbf{B}_{S_2, S_2}\|_{(k)}\},$$

where the positive integer $k \leq |S_1|$.

Proof: When $\mathbf{B} \in \mathcal{S}_+^n$ and $|S_1| = |S_2| = s$, for any optimal solution $(\mathbf{U}^*, \mathbf{V}^*)$ to problem (12) with $\mathbf{A} := \mathbf{B}$, according to the inequality in the proof of Proposition 3, we show

$$\begin{aligned} \|\mathbf{B}_{S_1, S_2}\|_{(k)} &= \text{tr}((\mathbf{U}^*)^\top \mathbf{B} \mathbf{V}^*) \leq \max\{\text{tr}((\mathbf{U}^*)^\top \mathbf{B} \mathbf{U}^*), \text{tr}((\mathbf{V}^*)^\top \mathbf{B} \mathbf{V}^*)\} \\ &\leq \max\{\|\mathbf{B}_{S_1, S_1}\|_{(k)}, \|\mathbf{B}_{S_2, S_2}\|_{(k)}\}, \end{aligned}$$

where the last inequality is because matrices \mathbf{U}^* and \mathbf{V}^* are feasible to problems

$$\begin{aligned} \max_{\mathbf{U} \in \mathbb{R}^{n \times k}} \{ \text{tr}(\mathbf{U}^\top \mathbf{B} \mathbf{U}) : \mathbf{U}^\top \mathbf{U} = \mathbf{I}_k, \mathbf{U}_{[n] \setminus S_1, [k]} = \mathbf{0}_{n-s, k} \} &= \|\mathbf{B}_{S_1, S_1}\|_{(k)}, \\ \max_{\mathbf{V} \in \mathbb{R}^{n \times k}} \{ \text{tr}(\mathbf{V}^\top \mathbf{B} \mathbf{V}) : \mathbf{V}^\top \mathbf{V} = \mathbf{I}_k, \mathbf{V}_{[n] \setminus S_2, [k]} = \mathbf{0}_{n-s, k} \} &= \|\mathbf{B}_{S_2, S_2}\|_{(k)}, \end{aligned}$$

where the equalities are due to theorem 23.2 in [40]. \square

In the selection Algorithm 1-Algorithm 3 for SSVD (4), the output is a pair of subsets (\hat{S}_1, \hat{S}_2) , while SPCA (13) only involves one subset. Given $\mathbf{A} \in \mathcal{S}_+^n$ and $s_1 = s_2 = s$, according to Lemma 3, we can adapt Algorithm 1-Algorithm 3 to solve SPCA (13) by comparing the objective values of $\mathbf{A}_{\hat{S}_1, \hat{S}_1}$ and $\mathbf{A}_{\hat{S}_2, \hat{S}_2}$ and outputting a better solution of \hat{S}_1 and \hat{S}_2 . In particular, since matrix \mathbf{A} is symmetric and positive semidefinite in SPCA (13), we can simplify the implementation of the selection Algorithm 2 by using one loop on rows only; and the third selection Algorithm 3 can also be simplified to one loop only for solving SPCA (13) since the top left- and right- singular vectors of positive semidefinite matrices are the same. The details of the selection algorithms for SPCA (13) can be found in Algorithm 6, Algorithm 7, and Algorithm 8. Their corresponding approximation ratios and time complexities for SPCA (13) are summarized in Corollary 4.

Algorithm 6 Selection Algorithm for SPCA (13)

- 1: **Input:** A matrix $\mathbf{A} \in \mathcal{S}_+^n$ and integers $s_1 = s_2 = s \in [n]$
- 2: Suppose (\hat{S}_1, \hat{S}_2) solves the problem below

$$\max_{S_1 \subseteq [m], S_2 \subseteq [n]} \{ \|\mathbf{A}_{S_1, S_2}\|_F : |S_1| \leq s_1, |S_2| \leq s_2 \}.$$

- 3: **Output:** A better solution of SPCA (13) between \hat{S}_1 and \hat{S}_2
-

Algorithm 7 Selection Algorithm for SPCA (13)

- 1: **Input:** A matrix $\mathbf{A} \in \mathcal{S}_+^n$, integers $s_1 = s_2 = s \in [n]$, and $1 \leq k \leq s$
 - 2: **for** $j \in [n]$ **do**
 - 3: Let \hat{S}_1^j denote the index set of the s_1 largest absolute entries in $\mathbf{A}_{:,j}$
 - 4: $\hat{S}_2^j \in \text{argmax}_{S \subseteq [n]} \{ \sum_{\ell \in S} \|\mathbf{A}_{\hat{S}_1^j, \ell}\|_2 : |S| = s_2 \}$
 - 5: **end for**
 - 6: Let (\hat{S}_1, \hat{S}_2) denote the best solution for SSVD (4) among the candidate set $\{(\hat{S}_1^j, \hat{S}_2^j)\}_{j \in [n]}$
 - 7: **Output:** A better solution of SPCA (13) between \hat{S}_1 and \hat{S}_2
-

Corollary 4 *The selection Algorithm 6-Algorithm 8 for SPCA (13) admit $1/\sqrt{s}$, $1/\sqrt{ks}$, and $s/(kn)$ approximation ratios, respectively, and all the ratios are tight when $s \leq n/2$; Moreover, the selection Algorithm 7 and Algorithm 8 admit $O(n^2 \log(n) + nks^2)$ and $O(n \log(n) + n^2)$ time complexities, respectively.*

Proof: See Appendix A.3. \square

Algorithm 8 Selection Algorithm for SPCA (13)

- 1: **Input:** A matrix $\mathbf{A} \in \mathcal{S}_+^n$, integers $s_1 = s_2 = s \in [n]$, and $1 \leq k \leq s$
 - 2: Compute the eigenvector \mathbf{u}_1 corresponding to the largest eigenvalue of \mathbf{A}
 - 3: Denote by \hat{S}_1 the indices of s_1 largest absolute entries in \mathbf{u}_1
 - 4: Denote by \hat{S}_2 the indices of s_2 largest absolute entries in $(\mathbf{u}_1)_{\hat{S}_1}^\top \mathbf{A}_{\hat{S}_1, [n]}$
 - 5: **Output:** A better solution of SPCA (13) between \hat{S}_1 and \hat{S}_2
-

5.3 Searching Algorithms of SPCA

The positive semidefiniteness of matrix \mathbf{A} allows us to simplify the implementations and improves theoretical performances of the greedy and local search algorithms for SPCA (13) compared to SSVD (4). The detailed procedures are presented in Algorithm 9 and Algorithm 10. It is worthy of mentioning that at the ℓ th iteration of the greedy Algorithm 9, if $\ell \leq k$, Step 4 reduces to

$$i^* \in \operatorname{argmax}_{i \in [n] \setminus \hat{S}} \operatorname{tr}(\mathbf{A}_{\hat{S} \cup \{i\}, \hat{S} \cup \{i\}}) = \operatorname{argmax}_{i \in [n] \setminus \hat{S}} A_{ii}.$$

Thus, at the first k iterations, the greedy Algorithm 9 selects the k largest diagonal elements in matrix \mathbf{A} .

Algorithm 9 Greedy Algorithm for SPCA (13)

- 1: **Input:** A matrix $\mathbf{A} \in \mathcal{S}_+^n$, integers $s \in [n]$, and $1 \leq k \leq s$
 - 2: Define the subset $\hat{S} := \emptyset$
 - 3: **for** $\ell \in [s]$ **do**
 - 4: $i^* \in \operatorname{argmax}_{i \in [n] \setminus \hat{S}} \|\mathbf{A}_{\hat{S} \cup \{i\}, \hat{S} \cup \{i\}}\|_{(\min\{\ell, k\})}$
 - 5: Update $\hat{S} := \hat{S} \cup \{i^*\}$
 - 6: **end for**
 - 7: **Output:** \hat{S}
-

Algorithm 10 Local Search Algorithm for SPCA (13)

- 1: **Input:** A matrix $\mathbf{A} \in \mathcal{S}_+^n$, integers $s \in [n]$, and $1 \leq k \leq s$
 - 2: Initialize \hat{S} as the output of greedy Algorithm 9
 - 3: **do**
 - 4: **for** each pair $(i, j) \in \hat{S} \times ([n] \setminus \hat{S})$ **do**
 - 5: **if** $\|\mathbf{A}_{\hat{S} \cup \{j\} \setminus \{i\}, \hat{S} \cup \{j\} \setminus \{i\}}\|_{(k)} > \|\mathbf{A}_{\hat{S}, \hat{S}}\|_{(k)}$ **then**
 - 6: Update $\hat{S} := \hat{S} \cup \{j\} \setminus \{i\}$
 - 7: **end if**
 - 8: **end for**
 - 9: **while** there is still an improvement
 - 10: **Output:** \hat{S}
-

We also derive the approximation ratios and time complexities of greedy Algorithm 9 and local search Algorithm 10 for SPCA.

Corollary 5 *The greedy Algorithm 9 and local search Algorithm 10 for solving SPCA (13) admit a k/s -approximation ratio and the ratio is tight for both algorithms when $s \leq n/2$. Moreover, the greedy Algorithm 9 and local search Algorithm 10 have time complexities of $O(nks^3)$ and $O(nks^3L/\delta)$, respectively.*

Proof: Let us first analyze the approximation ratio of greedy Algorithm 9. To begin with, let T denote the index set of k largest diagonal elements in matrix \mathbf{A} . Suppose that an optimal solution to SPCA (13) is set S^* and the output of greedy Algorithm 9 is \hat{S} , then we have

$$w^{spca} = \|\mathbf{A}_{S^*, S^*}\|_{(k)} = \sum_{i \in [k]} \lambda_i(\mathbf{A}_{S^*, S^*}) \leq \text{tr}(\mathbf{A}_{S^*, S^*}) = \sum_{i \in S^*} A_{ii} \leq \frac{k}{s} \sum_{i \in T} A_{ii} \leq \frac{k}{s} \|\mathbf{A}_{\hat{S}, \hat{S}}\|_{(k)},$$

where the last inequality is because at the first k iterations, greedy Algorithm 9 picks the k largest diagonal elements of matrix \mathbf{A} . Thus, the greedy Algorithm 9 yields the k/s -approximation ratio for SPCA (13). The ratio is tight since the following example can achieve it.

Example 5 For any positive integers $s \geq k > 0$, let $n = 2s$ and matrix $\mathbf{A} \in \mathcal{S}_+^n$ be

$$\mathbf{A} = \sum_{i \in [s]} \mathbf{e}_i \mathbf{e}_i^\top + \sum_{i \in [k]} \begin{pmatrix} \mathbf{0}_s \\ \mathbf{1}_s \end{pmatrix} \begin{pmatrix} \mathbf{0}_s \\ \mathbf{1}_s \end{pmatrix}^\top = \begin{pmatrix} \mathbf{I}_s & \mathbf{0}_{s,s} \\ \mathbf{0}_{s,s} & \mathbf{1}_{s,s} \end{pmatrix},$$

where for each $i \in [s]$, vector $\mathbf{e}_i \in \mathbb{R}^n$ denotes the i th column of the identity matrix \mathbf{I}_n .

In Example 5, the greedy Algorithm 9 outputs the solution $\hat{S} = [s]$ with the objective value of SPCA (13) equal to k . However, the true optimal value SPCA (13) is $\|\mathbf{A}_{[s+1,n], [s+1,n]}\|_{(k)} = s$, which thus indicates that the k/s -approximation ratio is tight when $s \leq n/2$.

Now, let us focus on deriving the approximation ratio of the local search Algorithm 10. As an improved heuristic method, the local search Algorithm 10 produces at least k/s -approximation ratio. Furthermore, in Example 5, the local search Algorithm 10 cannot improve the solution $\hat{S} = [s]$ from the greedy Algorithm 9 and thus the k/s -approximation ratio is also tight.

Plugging $m = n$ and $s_1 = s_2$ into the time complexities for greedy Algorithm 4 and local search Algorithm 5 of SSVD, we obtain the corresponding time complexities for SPCA. \square

6 Branch-and-Cut Algorithms for Solving SPCA and SSVD to Optimality

In this section, we develop branch-and-cut algorithms with closed-form cuts to solve SPCA (5) and SSVD (4), respectively.

6.1 Branch-and Cut-Algorithm for SPCA

The branch-and-cut algorithm is built on an equivalent MILP formulation of SPCA as below.

Theorem 7 *The SPCA (16) is equivalent to*

$$(\text{SPCA}) \quad w^{spca} := \max_{\mathbf{z} \in \{0,1\}^n, w \in \mathbb{R}} \left\{ w : \sum_{i \in [n]} z_i \leq s, w \leq \|\mathbf{A}_{S,S}\|_{(k)} + \sum_{i \in [n] \setminus S} A_{ii} z_i, \forall S \subseteq [n], |S| \leq s \right\}. \quad (17)$$

Proof: See Appendix A.4. \square

Note that SPCA (17) is amenable to the branch-and-cut method. That is, we start with a relaxed master problem with a small subset of constraints in SPCA (17) and given an incumbent solution $\hat{\mathbf{z}}$ with its support \hat{S} , we add the most violated constraint (i.e., $w \leq \|\mathbf{A}_{\hat{S}, \hat{S}}\|_{(k)} + \sum_{i \in [n] \setminus \hat{S}} A_{ii} \hat{z}_i$) into the relaxed master problem and then repeat the solution procedure until the objective value of the relaxed problem is equal to the best lower bound. This solution procedure will terminate after a finite number of iterations. To be specific, given a pair of solutions $(\hat{w}, \hat{\mathbf{z}})$ of SPCA (17), the most violated constraint results from the fact that the set \hat{S} achieves the maximum gap between \hat{w} and $\|\mathbf{A}_{S,S}\|_{(k)} + \sum_{i \in [n] \setminus S} A_{ii} \hat{z}_i$ among all candidate sets S of size up to s , as summarized follows.

Corollary 6 For a pair of solutions (\hat{w}, \hat{z}) of SPCA (17), suppose that \hat{S} denotes the support of \hat{z} , then we have

$$\max_{S \subseteq [n], |S| \leq s} \left\{ \hat{w} - \|\mathbf{A}_{S,S}\|_{(k)} - \sum_{i \in [n] \setminus S} A_{ii} \hat{z}_i \right\} = \hat{w} - \|\mathbf{A}_{\hat{S}, \hat{S}}\|_{(k)} - \sum_{i \in [n] \setminus \hat{S}} A_{ii} \hat{z}_i.$$

Proof: See Appendix A.5. \square

6.2 Branch-and-Cut Algorithm for SSVD

Using the Cholesky decomposition of matrix \mathbf{A} , SPCA (5) possesses an MISDP (16), which enables us to develop a branch-and-cut algorithm. However, this is difficult to directly extend to SSVD (4) since the matrix \mathbf{A} may not even be symmetric, nor does its Cholesky decomposition exist. Fortunately, the following key lemma helps address this difficulty.

Lemma 4 For any matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$, let us define its augmented version $\overline{\mathbf{B}} \in \mathbb{R}^{m+n, m+n}$ as below

$$\overline{\mathbf{B}} \in \mathbb{R}^{m+n, m+n} := \begin{pmatrix} \mathbf{0}_{m,m} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{0}_{n,n} \end{pmatrix} + t \mathbf{I}_{m+n},$$

for some constant $t \in \mathbb{R}$. Then we have

(i) The eigenvalues of matrix \mathbf{B} are equal to

$$\lambda_i(\overline{\mathbf{B}}) = \begin{cases} t + \sigma_i(\mathbf{B}), & \forall i \in [r], \\ t, & \forall i \in [r+1, m+n-r], \\ t - \sigma_{m+n+1-i}(\mathbf{B}), & \forall i \in [m+n-r+1, m+n]; \end{cases}$$

(ii) The matrix $\overline{\mathbf{B}}$ is positive semidefinite if $t \geq \sigma_1(\mathbf{B})$.

Proof: Part (i) directly follows from the result in Chapter 3.2. of [1].

Part (ii) stems from the facts that matrix $\overline{\mathbf{B}}$ is symmetric and the smallest eigenvalue is $t - \sigma_1(\mathbf{B}) \geq 0$.

\square

By constructing the augmented counterpart of matrix \mathbf{A} , we can derive another equivalent combinatorial formulation of SSVD (4) as below.

Proposition 4 The SSVD (4) is equivalent to

$$(\text{SSVD}) \quad w^* := \max_{S \subseteq [m+n]} \left\{ \|\overline{\mathbf{A}}_{S,S}\|_{(k)} : |S \cap [m]| \leq s_1, |S \cap [m+1, m+n]| \leq s_2 \right\} - k\sigma_1(\mathbf{A}), \quad (18)$$

where matrix $\overline{\mathbf{A}} \in \mathcal{S}_+^{m+n}$ is defined by

$$\overline{\mathbf{A}} \in \mathcal{S}_+^{m+n} := \begin{pmatrix} \mathbf{0}_{m,m} & \mathbf{A} \\ \mathbf{A}^\top & \mathbf{0}_{n,n} \end{pmatrix} + \sigma_1(\mathbf{A}) \mathbf{I}_{m+n}.$$

Proof: See Appendix A.6. \square

According to Part (ii) in Lemma 4, matrix $\overline{\mathbf{A}}$ defined in Proposition 4 is positive semidefinite. Let $\overline{\mathbf{A}} = \overline{\mathbf{C}}^\top \overline{\mathbf{C}}$ denote its Cholesky decomposition, where $\overline{\mathbf{C}} \in \mathbb{R}^{\bar{r} \times (m+n)}$ is the corresponding Cholesky matrix, \bar{r} is the rank of matrix $\overline{\mathbf{A}}$ and for each $i \in [m+n]$, let $\overline{\mathbf{c}}_i \in \mathbb{R}^{\bar{r}}$ denote its i th column. With the notation introduced, a second MISDP formulation of SSVD (18) can be derived as below.

Corollary 7 *The SSVD (18) is equivalent to*

$$(\text{SSVD}) \quad w^* := \max_{\substack{\mathbf{z} \in \{0,1\}^{m+n}, \mathbf{X} \in \mathcal{S}_+^{\overline{\mathbf{r}}}, \\ \mathbf{W}_1, \dots, \mathbf{W}_{m+n} \in \mathcal{S}_+^{\overline{\mathbf{r}}}}} \left\{ \sum_{i \in [m+n]} \bar{\mathbf{c}}_i^\top \mathbf{W}_i \bar{\mathbf{c}}_i : \mathbf{X} \preceq \mathbf{I}_{\overline{\mathbf{r}}}, \text{tr}(\mathbf{X}) \leq k, \mathbf{X} \succeq \mathbf{W}_i, \text{tr}(\mathbf{W}_i) \leq z_i, \right. \\ \left. \forall i \in [m+n], \sum_{i \in [m]} z_i \leq s_1, \sum_{i \in [m+1, m+n]} z_i \leq s_2 \right\} - k\sigma_1(\mathbf{A}). \quad (19)$$

Proof: The proof is almost identical to that of Corollary 3 and is thus omitted. \square

Based on the MISDP (19), we can also derive an MILP formulation of SSVD, which is the key to designing a branch-and-cut algorithm.

Corollary 8 *The SSVD (19) is equivalent to*

$$(\text{SSVD}) \quad w^* := \max_{\mathbf{z} \in \{0,1\}^{m+n}, w} \left\{ w : \sum_{i \in [m]} z_i \leq s_1, \sum_{i \in [m+1, m+n]} z_i \leq s_2, w \leq \|\bar{\mathbf{A}}_{S,S}\|_{(k)} + \sum_{i \in [m+n] \setminus S} \bar{A}_{ii} z_i, \forall S \subseteq [m+n], |[m] \cap S| \leq s_1, |[m+1, m+n] \cap S| \leq s_1 \right\} - k\sigma_1(\mathbf{A}). \quad (20)$$

Proof: The proof is almost identical to that of Theorem 7 and is thus omitted. \square

Similarly, we can apply the branch-and-cut method to solve SSVD (20). That is, we start with a master problem with a small subset of constraints in SSVD (20) and given an incumbent solution $\hat{\mathbf{z}}$ with support \hat{S} , we add the most violated constraint (i.e., $w \leq \|\bar{\mathbf{A}}_{\hat{S}, \hat{S}}\|_{(k)} + \sum_{i \in [m+n] \setminus \hat{S}} \bar{A}_{ii} z_i$), into the relaxed master problem and then repeat the solution procedure until the objective value of the relaxed problem is equal to the best lower bound.

7 Numerical Results

In this section, we implement our proposed exact and approximation algorithms to solve synthetic and real-world instances of SSVD (4) and SPCA (5). Specifically, we consider four real-world datasets from in [11], a drug abuse dataset from the National Survey on Drug Use and Health in 2018, and a solar flare dataset in [51], which are described in Table 3.

Table 3: Statistics of real datasets

DATASET	m	n	RANK	POSITIVE SEMIDEFINITE
EISEN-1	79	79	79	YES
EISEN-2	118	118	78	YES
COLON	500	500	62	YES
REDDIT	2000	2000	949	YES
DRUG	16313	2365	2224	NO
MOUSSE	232	292	232	NO

All the experiments are conducted in Python 3.6 with calls to Gurobi 9.0 on a personal PC with 2.3 GHz Intel Core i5 processor and 8G of memory. Note that the Ky Fan k -norm of matrices is efficiently computed by the PRIMME package [49].

7.1 Illustration of Sparsity Levels on Synthetic and Real Datasets

This subsection illustrates how our SSVD model (4) performs against the sparsity levels (i.e., the tuning parameters (s_1, s_2)) using synthetic and real-world data. Particularly, the synthetic data matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ are obtained by sampling $m \times n$ i.i.d. random standard normal variables.

To evaluate SSVD (4) and illustrate the selection of sparsity levels, we compute the ratio of its optimal value over the Ky Fan k -norm of the original data matrix \mathbf{A} , i.e., $w^*/\|\mathbf{A}\|_{(k)}$ for various parameters s_1, s_2 . Here, the larger the ratio means that the more information that the optimal submatrix from SSVD contains. Using the MILP formulation (8) of SSVD, we develop a branch-and-cut algorithm to exactly compute the optimal values for the synthetic dataset with $(m, n, k) = (8, 10, 3)$. Figure 2a presents how the ratio $w^*/\|\mathbf{A}\|_{(k)}$ varies as s_1 and s_2 change. Note that the running time of all the cases is less than one minute.

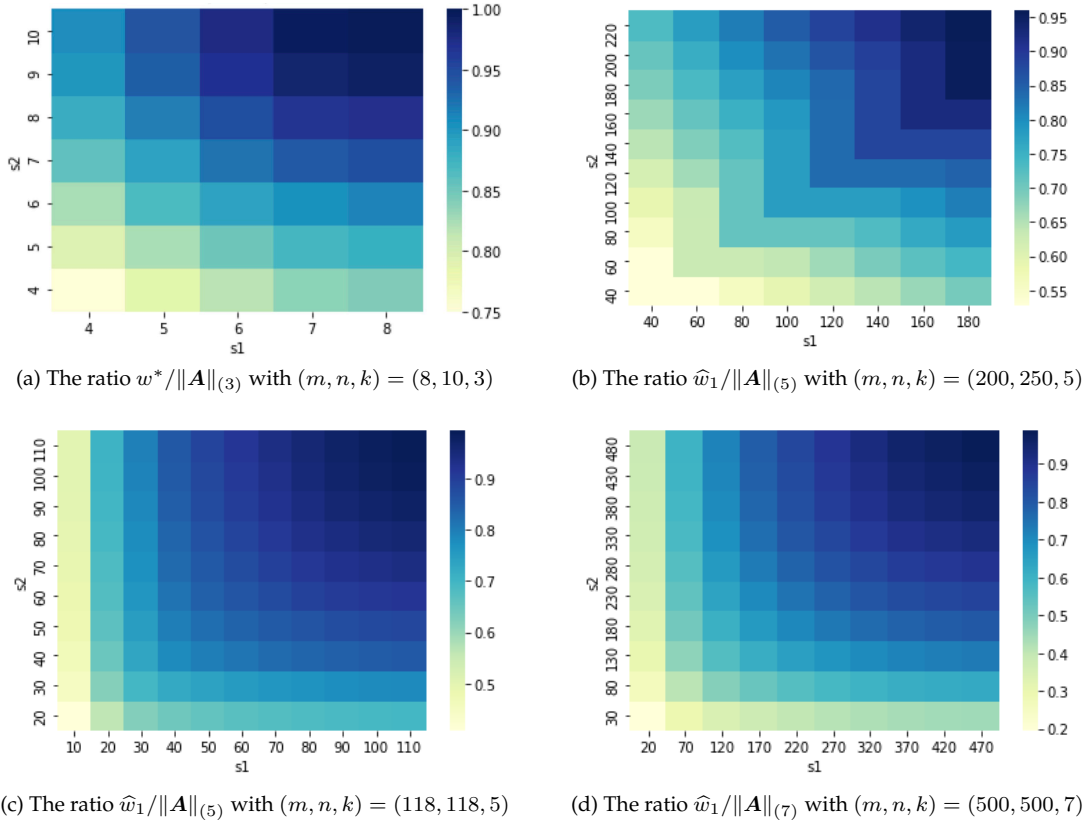


Fig. 2: Illustration of sparsity levels

We also test the larger instances, where we employ the scalable approximation Algorithm 2, Algorithm 3, and Algorithm 4, and return the best objective value as the lower bound of SSVD (4), denoted by \hat{w}_1 . Accordingly, we compute the approximate ratio by $\hat{w}_1/\|\mathbf{A}\|_{(k)}$. Figure 2b illustrates the ratio against parameters s_1 and s_2 for the synthetic instance of $m = 200$, $n = 250$, and $k = 5$. In addition, we test Eisen-2 and Colon datasets in Table 3 with $k = 5$ and $k = 7$, respectively, as displayed in Figure 2c and Figure 2d. Note that the running time of approximation algorithms on all the cases is also less than one minute. For the two synthetic and real datasets in Figure 2, we can observe that the optimal submatrix from SSVD (4) captures about 80% of the information in matrix \mathbf{A} when $s_1 \approx m/2$ and $s_2 \approx n/2$, i.e., a quarter size of the original matrix \mathbf{A} .

7.2 Evaluations of the Algorithms on Real Datasets

In this subsection, we evaluate our proposed exact and approximation algorithms on

We apply the proposed exact and approximation algorithms to solving SPCA (5) on the first three positive semidefinite matrices with various parameters s and k . Table 4 presents the numerical results, where “Paras” denotes the parameters, “B&C” denotes the branch-and-cut algorithm, “Alg” denotes the algorithm, the optimality gap(%) is computed by

$$\text{gap}(\%) = 100 \times \frac{\text{optimal value} - \text{lower bound}}{\text{lower bound}},$$

“LS” denotes the benchmark, a modified local search algorithm proposed by [12], which uses a random initial solution, and “Ran” and “Vol” denote the randomized leverage-score algorithm in [26] and maximum volume-based algorithm in [29] for submatrix selection, respectively. Note that the computational time of the approximation Algorithm 7-Algorithm 10 are at most one second and thus omitted. It can be seen that the branch-and-cut method is able to solve small- or medium-sized instances to optimality, which allows us to test the solution quality of the approximation algorithms by computing their optimality gaps as shown in Table 4. It is worth mentioning that our selection Algorithm 7 and local search Algorithm 10 are consistently better than the benchmark. We also see that greedy Algorithm 9 and local search Algorithm 10 returns the optimal solution with zero optimality gap for most of the instances. Lastly, it is interesting to note that the solution quality of the truncation Algorithm 6 is more sensitive to the parameter s than k , which complies with its theoretical approximation ratio.

Table 4: Exact and approximation algorithms for solving SPCA (5)

DATA	PARAS		B&C		GAP(%) OF APPROXIMATION ALGORITHMS								
	s	k	w^{spca}	TIME(S)	ALG 6	TIME(S)	ALG 7	ALG 8	ALG 9	ALG 10	LS	RAN	VOL
EISEN-1 $n = 79$	5	2	15.9771	1	0.00	1	0.00	0.00	0.00	0.00	0.00	35	40
	5	3	16.1501	1	0.00	1	0.00	0.00	0.00	0.00	0.00	18	40
	5	4	16.2399	1	0.00	1	0.00	0.00	0.00	0.00	0.00	15	39
	10	2	20.1578	1	0.00	1	0.00	2.50	0.00	0.00	4.58	53	16
	10	3	20.5690	1	0.00	1	0.00	2.30	0.00	0.00	4.03	44	14
	10	4	20.8163	1	0.00	1	0.00	2.30	0.00	0.00	3.85	33	14
	15	2	20.7619	180	0.96	1	0.33	3.65	0.00	0.00	6.80	29	2
	15	3	22.2571	1	3.25	1	0.00	5.92	0.00	0.00	10.84	25	5
	15	4	22.7143	1	2.55	1	0.07	5.14	0.00	0.00	10.19	33	5
EISEN-2 $n = 118$	5	2	8.1438	6	0.00	1	0.00	19.75	14.45	0.93	19.75	30	35
	5	3	8.4338	10	2.08	1	2.06	22.70	0.00	0.00	22.70	28	32
	5	4	8.5467	10	2.87	1	2.87	23.57	0.00	0.00	23.57	8	31
	10	2	13.8929	510	15.31	1	0.83	15.31	7.71	1.45	6.30	7	29
	10	3	14.4792	2095	18.21	1	0.15	17.83	0.00	0.00	8.51	65	18
	10	4	14.7313	3325	18.56	1	0.00	17.99	0.00	0.00	9.12	51	17
COLON $n = 500$	5	2	1709.9575	10	0.00	302	0.00	0.00	0.00	0.00	7.55	169	83
	5	3	1718.1590	25	0.00	297	0.00	0.00	0.00	0.00	5.85	56	77
	5	4	1720.1219	20	0.00	303	0.00	0.00	0.00	0.00	5.60	115	72
	10	2	2794.1404	3951	0.38	315	0.00	0.38	0.00	0.00	0.21	98	114

We further employ the proposed approximation algorithms to solve SSVD (3) on all data matrices by selecting different parameters s_1 and s_2 . The computational results can be found in Table 5, where “LB” denotes the lower bound returned by an approximation algorithm. Note that for these SSVD instances, the algorithm developed by [12] cannot solve them since the selected submatrices might not be symmetric.

In Table 4, we see that the other two existing algorithms perform poorly even when solving the small instances. We also notice that the branch-and-cut algorithm has difficulty in finding a feasible solution within one hour. Thus, neither the benchmark algorithm nor the branch-and-cut algorithm is reported. It is worthy of mentioning that to more efficiently solve high-dimensional Drug dataset, we restrict the selection Algorithm 1 and local search Algorithm 5 to a proper but larger-sized submatrix than $s_1 \times s_2$ by first greedily selecting a larger-sized (e.g., $2s_1 \times 2s_2$) submatrix and then applying these two algorithms to the submatrix. To better illustrate the comparison of the approximation algorithms for SSVD (3), we normalize the lower bounds by the ones found by the local search Algorithm 5 and display the results in Figure 3. A larger ratio means a larger lower bound and a better solution. We see that the local search Algorithm 5 significantly improves the solution of greedy Algorithm 4 and consistently performs the best among all algorithms.

Table 5: Approximation algorithms for solving SSVD (4)

DATA	PARAS s_1 s_2 k	ALG 1		ALG 2		ALG 3		ALG 4		ALG 5	
		LB	TIME(S)	LB	TIME(S)	LB	TIME(S)	LB	TIME(S)	LB	TIME(S)
EISEN-1	5 10 3	17.89	1	17.89	1	17.89	1	16.15	1	18.08	1
$m = 79$	10 15 3	20.82	1	20.97	1	20.32	1	20.57	1	21.11	2
$n = 79$	15 20 3	22.15	1	21.78	1	21.12	1	22.26	1	22.53	1
EISEN-2	15 20 4	21.38	1	21.38	1	19.60	1	19.39	1	21.41	4
$m = 118$	20 25 4	25.24	1	25.24	1	25.24	1	23.89	2	25.32	3
$n = 118$	25 30 4	28.02	1	28.09	1	28.03	1	26.96	2	28.83	8
COLON	30 25 5	5943.40	226	5957.77	1	5943.40	1	5621.61	13	5987.91	57
$m = 500$	35 30 5	6712.53	209	6714.23	1	6712.53	1	6391.81	16	6724.51	23
$n = 500$	40 35 5	7387.09	280	7387.08	1	7387.08	1	7087.79	19	7405.59	27
REDDIT	40 35 6	4260.76	371	4230.60	6	3761.87	1	4273.96	92	4364.84	403
$m = 2000$	45 40 6	4634.98	340	4409.92	6	3882.21	1	4461.45	104	4654.04	1052
$n = 2000$	50 45 6	4786.25	363	4601.39	8	3963.54	1	4734.31	132	4809.57	922
DRUG	30 35 7	827.21	554	821.38	203	766.41	1	982.74	163	1012.91	549
$m = 16313$	35 30 7	873.98	722	862.99	173	808.94	1	982.74	236	1067.52	724
$n = 2365$	50 45 7	1070.86	1442	1064.35	253	997.17	1	1247.86	412	1327.95	1332
	45 50 7	1034.86	1205	1036.71	271	961.55	1	1247.86	331	1286.00	1121

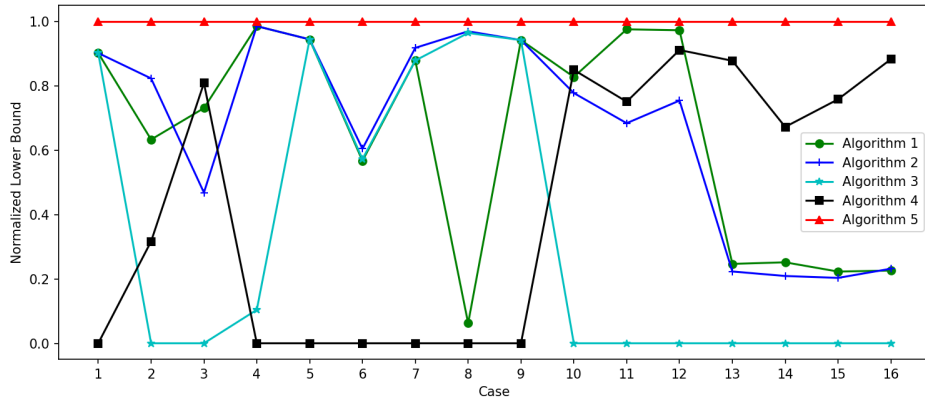


Fig. 3: Normalized lower bounds in Table 5

7.3 Anomaly Detection by SSVD: Solar Flare Detection

Recently, the low-rank-based methods [6, 38] or low-rank-embedded deep learning methods [19, 43] have

been widely applied to anomaly detection, especially in hyperspectral imagery [55], where the basic idea is that the normal data can be recast in a low-dimensional subspace and admit low-rank property after removing the anomalies. Since our SSVD (3) achieves simultaneously sparse and low-rank approximation, it can be directly applied to the solar flare detection. Particularly, the solar flare phenomenon, a rapid and intense variation of brightness in the solar atmosphere lasting from minutes to hours, may potentially pose risks to electronic power grids and radio communications and thus needs efficiently detecting.

The solar dataset-MOUSSE in Table 3 used for the anomaly detection of SSVD (3) consists of 300 solar images in a video format collected from a satellite, where each image is of size 232×292 pixels [51]. Therefore, there are 300 temporal data matrices of equal size, 232×292 . Based on the previous numerical performance, we choose to adopt the truncation Algorithm 2 of SSVD to find a sparse and low-rank approximation for each matrix by setting $s_1 = 100$, $s_2 = 100$, and $k = 10$, where the output values of truncation Algorithm 2 representing the approximation results of matrices at all time points are displayed in Figure 4a. For comparison purposes, the sum of all the singular values of each data matrix is shown in Figure 4b without any sparse or low-rank approximation. Both methods can be implemented within several minutes. In the temporal solar dataset, two occurrences of solar flares at intervals $187 \sim 202$ and $216 \sim 268$ have been recognized by [33]. It is seen that the sparse and low-rank approximation by SSVD (3) successfully identifies the potential solar flares at red points $177 \sim 201$ and $217 \sim 267$ in Figure 4a, which is slightly different but consistent with the literature. On the other hand, we can only identify the solar flare occurrence at $217 \sim 236$ if we simply use the original data matrix for detection in Figure 4b.

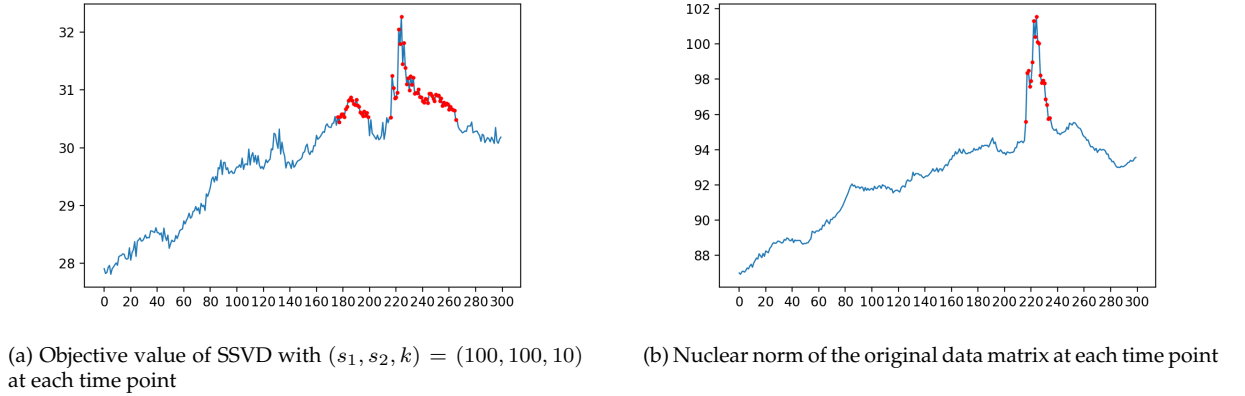


Fig. 4: Sparse and low-rank approximation of SSVD for solar flare detection

As seen in Figure 4, the proposed SSVD model is not only more effective but also improves interpretability and offers insights of the anomaly detection results. Figure 5 shows comparisons of original images and the selected sub-image by the SSVD at two time points, 192 and 222, respectively. We see that at both time points, the selected sub-images capture the abnormal brightness and remove a large portion of the irrelevant dark solar atmosphere. These selected sub-images can be useful for domain experts to pinpoint the important sub-image and study the flares more closely. This demonstrates that the SSVD can succeed in extracting valuable information and selecting crucial features.

8 Conclusions

For the best submatrix selection within a given low rank, we present a novel SSVD model, which ensures the selected submatrix to be simultaneously sparse and low-rank. Future interesting directions on SSVD include finding the optimal approximation ratio of a polynomial algorithm for SSVD, studying how to finalize the effective sparsity levels, and developing efficient algorithms for the variants of SSVD: robust SSVD un-

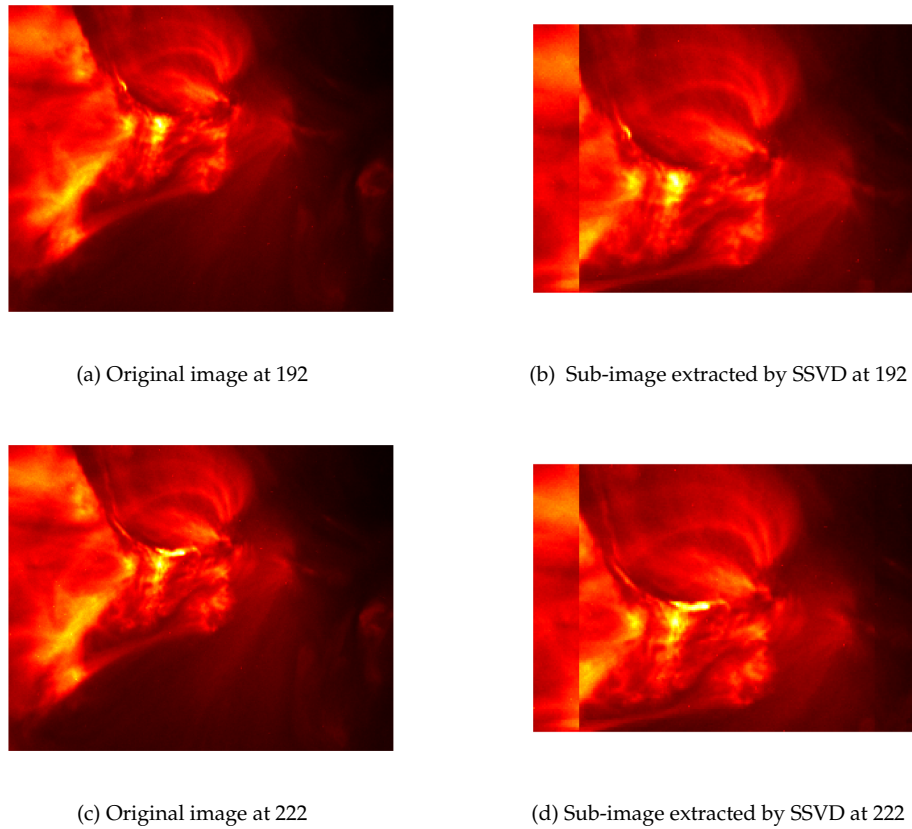


Fig. 5: Approximation of solar flare images by SSVD

der noisy data, fair SSVD among multiple sensitive groups, sparse and low-rank tensor decomposition, and so on.

Acknowledgements This research has been supported in part by the National Science Foundation grants 2246414 and 2246417 and Office of Naval Research grant N00014-24-1-2066. The authors would like to thank Dr. Hao Yan from Arizona State University for sharing the solar flare dataset.

References

1. A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001.
2. L. Berk and D. Bertsimas. Certifiably optimal sparse principal component analysis. *Mathematical Programming Computation*, 11(3):381–420, 2019.
3. C. Boutsidis and D. P. Woodruff. Optimal cur matrix decompositions. *SIAM Journal on Computing*, 46(2):543–589, 2017.
4. S. O. Chan, D. Papaillopoulos, and A. Rubinstein. On the approximability of sparse PCA. In *Conference on Learning Theory*, pages 623–646, 2016.
5. W. Chen. Simultaneously sparse and low-rank matrix reconstruction via nonconvex and nonseparable regularization. *IEEE transactions on signal processing*, 66(20):5313–5323, 2018.

6. T. Cheng and B. Wang. Graph and total variation regularized low-rank representation for hyperspectral anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing*, 58(1):391–406, 2019.
7. A. Chowdhury, P. Drineas, D. P. Woodruff, and S. Zhou. Approximation algorithms for sparse principal component analysis. *arXiv preprint arXiv:2006.12748*, 2020.
8. A. Civril and M. Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49):4801–4811, 2009.
9. A. Cortinovis and D. Kressner. Low-rank approximation in the frobenius norm by column and row subset selection. *SIAM Journal on Matrix Analysis and Applications*, 41(4):1651–1673, 2020.
10. A. Deshpande and L. Rademacher. Efficient volume sampling for row /column subset selection. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 329–338. IEEE, 2010.
11. S. S. Dey, R. Mazumder, and G. Wang. A convex integer programming approach for optimal sparse PCA. *arXiv preprint arXiv:1810.09062*, 2018.
12. S. S. Dey, M. Molinaro, and G. Wang. Solving row-sparse principal component analysis via convex integer programs. *arXiv preprint arXiv:2010.11152*, 2020.
13. X. V. Doan and S. Vavasis. Finding the largest low-rank clusters with Ky Fan 2-k-norm and ℓ_1 -norm. *SIAM Journal on Optimization*, 26(1):274–312, 2016.
14. T. Gally, M. E. Pfetsch, and S. Ulbrich. A framework for solving mixed-integer semidefinite programs. *Optimization Methods and Software*, 33(3):594–632, 2018.
15. C. Gao, Y. Lu, Z. Ma, and H. H. Zhou. Optimal estimation and completion of matrices with biclustering structures. *The Journal of Machine Learning Research*, 17(1):5602–5630, 2016.
16. G. H. Golub, A. Hoffman, and G. W. Stewart. A generalization of the eckart-young-mirsky matrix approximation theorem. *Linear Algebra and its applications*, 88:317–327, 1987.
17. G. H. Golub and C. F. Van Loan. *Matrix computations*. JHU press, 2013.
18. C. Jeong and X. Fang. Two-dimensional variable selection and its applications in the diagnostics of product quality defects. *IIE Transactions*, 54(7):619–629, 2022.
19. K. Jiang, W. Xie, J. Lei, T. Jiang, and Y. Li. Lren: Low-rank embedded network for sample-free hyperspectral anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4139–4146, 2021.
20. J. Kim, M. Tawarmalani, and J.-P. P. Richard. Convexification of permutation-invariant sets and an application to sparse principal component analysis. *Mathematics of Operations Research*, 2021.
21. C.-W. Ko, J. Lee, and M. Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.
22. M. Lee, H. Shen, J. Z. Huang, and J. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, 2010.
23. Y. Li and W. Xie. Exact and approximation algorithms for sparse PCA. *arXiv preprint arXiv:2008.12438*, 2020.
24. Y. Li and W. Xie. Best principal submatrix selection for the maximum entropy sampling problem: scalable algorithms and performance guarantees. *Operations Research*, 2023.
25. M. Magdon-Ismail. NP-hardness and inapproximability of sparse PCA. *Information Processing Letters*, 126:35–38, 2017.
26. M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.
27. I. Markovsky and K. Usevich. *Low rank approximation*, volume 139. Springer, 2012.
28. G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.
29. A. Mikhalev and I. V. Oseledets. Rectangular maximum-volume submatrices and their applications. *Linear Algebra and its Applications*, 538:187–211, 2018.
30. W. Min, J. Liu, F. Luo, and S. Zhang. A novel two-stage method for identifying microRNA-gene regulatory modules in breast cancer. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 151–156. IEEE, 2015.

31. W. Min, J. Liu, and S. Zhang. L0-norm sparse graph-regularized SVD for biclustering. *arXiv preprint arXiv:1603.06035*, 2016.
32. W. Min, J. Liu, and S. Zhang. Group-sparse SVD models and their applications in biological data. *arXiv preprint arXiv:1807.10956*, 2018.
33. M. Nabhan, Y. Mei, and J. Shi. Correlation-based dynamic sampling for online high dimensional process monitoring. *Journal of Quality Technology*, 53(3):289–308, 2021.
34. X. Niu, L. Fu, W. Zhang, and Y. Li. Seismic data interpolation based on simultaneously sparse and low-rank matrix recovery. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.
35. A. Omanović. *Knowledge-constrained projection of high-dimensional data*. PhD thesis, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2018.
36. A. Osinsky and N. L. Zamarashkin. Pseudo-skeleton approximations with better accuracy estimates. *Linear Algebra and its Applications*, 537:221–249, 2018.
37. J. Y. Park, K. Carr, S. Zheng, Y. Yue, and R. Yu. Multiresolution tensor learning for efficient and interpretable spatial analysis. In *International Conference on Machine Learning*, pages 7499–7509. PMLR, 2020.
38. Y. Qu, W. Wang, R. Guo, B. Ayhan, C. Kwan, S. Vance, and H. Qi. Hyperspectral anomaly detection through spectral unmixing and dictionary-based low-rank decomposition. *IEEE Transactions on Geoscience and Remote Sensing*, 56(8):4391–4405, 2018.
39. J. Seberry, B. J. Wysocki, and T. A. Wysocki. On some applications of hadamard matrices. *Metrika*, 62(2-3):221–239, 2005.
40. S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
41. S. L. Shishkin, A. Shalaginov, and S. D. Bopardikar. Fast approximate truncated SVD. *Numerical Linear Algebra with Applications*, 26(4):e2246, 2019.
42. M. Sill, S. Kaiser, A. Benner, and A. Kopp-Schneider. Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics*, 27(15):2089–2097, 2011.
43. S. Song, H. Zhou, Y. Yang, and J. Song. Hyperspectral anomaly detection via convolutional neural network and low rank with density-based clustering. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9):3637–3649, 2019.
44. E. Tyrtshnikov, S. Goreinov, and N. Zamarashkin. Pseudo-skeleton approximations. *Doklay Akademii Nauk*, 343(2):151–152, 1995.
45. V. Vu and J. Lei. Minimax rates of estimation for sparse PCA in high dimensions. In *Artificial intelligence and statistics*, pages 1278–1286, 2012.
46. G. Watson. On matrix approximation problems with k fan k norms. *Numerical Algorithms*, 5(5):263–272, 1993.
47. L. Wei, A. Gómez, and S. Küçükyavuz. Ideal formulations for constrained convex optimization problems with indicator variables. *Mathematical Programming*, pages 1–32, 2021.
48. D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.
49. L. Wu and A. Stathopoulos. Primme svds: A preconditioned svd solver for computing accurately singular triplets of large matrices based on the primme eigensolver. *arXiv preprint arXiv:1408.5535*, 2014.
50. D. Xia. Optimal Schatten- q and Ky-Fan- k norm rate of low rank matrix estimation. *arXiv preprint arXiv:1403.6499*, 2014.
51. H. Yan, K. Paynabar, and J. Shi. Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition. *Technometrics*, 60(2):181–197, 2018.
52. D. Yang, Z. Ma, and A. Buja. A sparse singular value decomposition method for high-dimensional data. *Journal of Computational and Graphical Statistics*, 23(4):923–942, 2014.
53. D. Yang, Z. Ma, and A. Buja. Rate optimal denoising of simultaneously sparse and low rank matrices. *The Journal of Machine Learning Research*, 17(1):3163–3189, 2016.
54. N. Zamarashkin. Pseudo-skeleton approximations by matrices of maximal volume. 1997.

-
55. X. Zhang, X. Ma, N. Huan, J. Gu, X. Tang, and L. Jiao. Spectral-difference low-rank representation learning for hyperspectral anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing*, 59(12):10364–10377, 2021.

Appendix A. Proofs

A.1 Proof of Theorem 1

Theorem 1 *The SSVD (3) can be formulated as*

$$\begin{aligned} (\text{SSVD}) \quad w^* := & \max_{\mathbf{z} \in \{0,1\}^m, \mathbf{y} \in \{0,1\}^n, \mathbf{X} \in \mathbb{R}^{n \times m}} \left\{ \text{tr}(\mathbf{A}\mathbf{X}) : \|\mathbf{X}\|_2 \leq 1, \|\mathbf{X}\|_* \leq k, \right. \\ & \left. \|\mathbf{X}_{:,i}\|_\infty \leq z_i, \forall i \in [m], \sum_{i \in [m]} z_i \leq s_1, \|\mathbf{X}_{j,:}\|_\infty \leq y_j, \forall j \in [n], \sum_{j \in [n]} y_j \leq s_2 \right\}. \end{aligned} \quad (7)$$

Proof: We prove the equivalence of SSVD (4) and problem (7) via the one-to-one solution correspondence.

First, for any feasible solution $(\mathbf{z}, \mathbf{y}, \mathbf{X})$ to problem (7), let us denote by S_1 and S_2 the supports of \mathbf{z} and \mathbf{y} , separately. Thus, (S_1, S_2) is feasible to SSVD (4) with the objective value $\|\mathbf{A}_{S_1, S_2}\|_{(k)}$. We will show that its corresponding objective value of problem (7) is also equal to $\|\mathbf{A}_{S_1, S_2}\|_{(k)}$ below.

According to the constraints in problem (7), we can derive that $X_{ij} = 0$, for all $([n] \times [m]) \setminus (S_2 \times S_1)$. Thus, problem (7) can reduce to the following maximization problem

$$\max_{\mathbf{X} \in \mathbb{R}^{|S_2| \times |S_1|}} \{ \text{tr}(\mathbf{A}_{S_1, S_2} \mathbf{X}_{S_2, S_1}) : \|\mathbf{X}_{S_2, S_1}\|_2 \leq 1, \|\mathbf{X}_{S_2, S_1}\|_* \leq k \}, \quad (21)$$

which has the optimal value $\|\mathbf{A}_{S_1, S_2}\|_{(k)}$ by Lemma 1.

Second, for any feasible solution (S_1, S_2) to SSVD (4), we let $z_i = 1$ if $i \in S_1$ and zero, otherwise for each $i \in [m]$ and let $y_j = 1$ if $j \in S_2$ and zero, otherwise for each $j \in [n]$. Next, according to Lemma 1, let us denote \mathbf{X}_{S_2, S_1} to be an optimal solution to problem (21) with the objective value equal to the Ky Fan k -norm of submatrix \mathbf{A}_{S_1, S_2} . We augment the matrix \mathbf{X}_{S_2, S_1} to be of size $m \times n$ by filling zero entries. Thus, the constructed solution $(\mathbf{z}, \mathbf{y}, \mathbf{X})$ is feasible to problem (7) with the same objective value $\|\mathbf{A}_{S_1, S_2}\|_{(k)}$.

In conclusion, problem (7) is an exact formulation of SSVD (4). \square

A.2 Proof of Proposition 1

Proposition 1 *The problem (8) is NP-hard and can be formulated as the following MILP*

$$\max_{\substack{\mathbf{z} \in \{0,1\}^{m \times n}, \\ \mathbf{x} \in \{0,1\}^m, \mathbf{y} \in \{0,1\}^n}} \left\{ \sum_{i \in [m]} \sum_{j \in [n]} A_{ij}^2 z_{ij} : z_{ij} \leq x_i, z_{ij} \leq y_j, \forall i \in [m], \forall j \in [n], \sum_{i \in [m]} x_i \leq s_1, \sum_{j \in [n]} y_j \leq s_2 \right\}.$$

Proof: We split the proof into two steps: NP-hardness and MILP reformulation.

(i) **NP-hardness.**

We show that problem (8) can be reduced to the well-known NP-hard maximum clique problem.

Given an undirected graph $G = (V, E)$, let $n = |V|$ denote the number of vertices. Then we construct matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ as below

$$A_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E \\ n^2, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}, \forall i \in [n], j \in [n].$$

We see that matrix \mathbf{A} is symmetric and diagonally dominant and is thus positive semidefinite.

Suppose $s_1 = s_2 = s \in [n]$. For any solution (S_1^*, S_2^*) to problem (8), if $S_1^* = S_2^*$, then s diagonal elements are chosen and the objective value is at least sn^2 . On the other hand, if $S_1^* \neq S_2^*$, at most $s-1$ diagonal elements are chosen and the objective value is at most $s(s-1) + (s-1)n^2 < sn^2$. Therefore,

letting (S_1^*, S_2^*) denote the optimal solution to problem (8), at optimality, we must have $S_1^* = S_2^* = S^*$. That is, problem (8) reduces to

$$\max_{S \subseteq [n]} \{ \|A_{S,S}\|_F : |S| \leq s \}. \quad (22a)$$

Next, we will show that finding the largest s such that the optimal value of problem (22a) is equal to $s(s-1) + sn^2$ is equivalent to finding the maximum clique in the graph $G = (V, E)$. Indeed, for each $s \in [n]$, if the optimal value of problem (22a) is equal to $s(s-1) + sn^2$, then we find a size- s clique with vertices indexed by S^* . The largest such s provides us a lower bound of the maximum clique of the graph $G = (V, E)$. On the other hand, given a size- s clique with its vertices indexed by S^* , clearly S^* is an optimal solution to problem (22a) with the optimal value $s(s-1) + sn^2$. Hence, the size of maximum clique in the graph $G = (V, E)$ is smaller than or equal to the largest s such that the optimal value of problem (22a) is equal to $s(s-1) + sn^2$.

(ii) **MILP reformulation.**

First, for any feasible subsets S_1 and S_2 to problem (8), the objective value is equal to

$$\|A_{S_1, S_2}\|_F = \sqrt{\sum_{i \in S_1} \sum_{j \in S_2} A_{ij}^2},$$

where the equality is because of the definition of Frobenius norm. Thus, by taking square of the objective function, problem (8) can be equivalently reformulated as

$$\max_{S_1 \subseteq [m], S_2 \subseteq [n]} \left\{ \sum_{i \in S_1} \sum_{j \in S_2} A_{ij}^2 : |S_1| \leq s_1, |S_2| \leq s_2 \right\}. \quad (22b)$$

Next, let us introduce binary variables $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^n$ such that $x_i=1$ if $i \in S_1$ and zero, otherwise for all $i \in [m]$ and $y_j=1$ if $j \in S_2$ and zero, otherwise for all $j \in [n]$. Problem (22b) can be recast as the following binary bilinear program

$$\max_{\mathbf{x} \in \{0,1\}^m, \mathbf{y} \in \{0,1\}^n} \left\{ \sum_{i \in [m]} \sum_{j \in [n]} A_{ij}^2 x_i y_j : \sum_{i \in [m]} x_i \leq s_1, \sum_{j \in [n]} y_j \leq s_2 \right\}.$$

Above, introducing binary variables $\mathbf{z} \in \{0,1\}^{m \times n}$ and using McCormick inequalities [28] to linearize the bilinear terms in the objective function, we arrive at the desirable MILP formulation.

□

A.3 Proof of Corollary 4

Corollary 4 *The selection Algorithm 6-Algorithm 8 for SPCA (13) admit $1/\sqrt{s}$, $1/\sqrt{ks}$, and $s/(kn)$ approximation ratios, respectively, and all the ratios are tight when $s \leq n/2$; Moreover, the selection Algorithm 7 and Algorithm 8 admit $O(n^2 \log(n) + nks^2)$ and $O(n \log(n) + n^2)$ time complexities, respectively.*

Proof: The proof includes two parts with respect to the approximation ratios and time complexities. Plugging $m = n$ and $s_1 = s_2 = s$ into the complexities of selection Algorithm 2 and Algorithm 3 for SSVD (4), we obtain the desired time complexities of Algorithm 7 and Algorithm 8.

In the following, we will focus on deriving the approximation ratios of the selection algorithms for SPCA (13), separately. When matrix $A \in S_+^n$ and $s_1 = s_2 = s$, we have the following results.

(i) **Selection Algorithm 6 for SPCA (13).**

Suppose that the output of selection Algorithm 6 is (\hat{S}_1, \hat{S}_2) .

$$w^{spca} = w^* \leq \frac{1}{\sqrt{\min\{s_1, s_2\}}} \|A_{\hat{S}_1, \hat{S}_2}\|_{(k)} \leq \frac{1}{\sqrt{s}} \max \left\{ \|A_{\hat{S}_1, \hat{S}_1}\|_{(k)}, \|A_{\hat{S}_2, \hat{S}_2}\|_{(k)} \right\},$$

where the first inequality is due to Theorem 2 and the second one is from Lemma 3.

Thus the modified selection algorithm yields the $1/\sqrt{s}$ -approximation ratio for SPCA (13). The ratio is tight since the following example can achieve it.

Example 6 Suppose that $k = s$, $n = 2s$, and matrix $A \in \mathcal{S}_+^n$ is defined by

$$A = \begin{pmatrix} \mathbf{1}_{s,s} & \mathbf{0}_{s,s} \\ \mathbf{0}_{s,s} & D \end{pmatrix}, D \in \mathcal{S}_+^s := \sum_{i \in [k]} \sqrt{s} e_i e_i^\top,$$

where $e_i \in \mathbb{R}^s$ is the i th column of the identity matrix I_s .

In Example 6, the selection Algorithm 6 returns a solution $\hat{S}_1 = [s]$ and $\hat{S}_2 = [s]$ since $\|\mathbf{1}_{s,s}\|_F = \|D\|_F$, and the resulting objective value of SPCA (13) is

$$\max \left\{ \|A_{\hat{S}_1, \hat{S}_1}\|_{(k)}, \|A_{\hat{S}_2, \hat{S}_2}\|_{(k)} \right\} = \|\mathbf{1}_{s,s}\|_{(k)} = s.$$

However, the true optimal value of SPCA (13) is $\|D\|_{(k)} = k\sqrt{s} = s\sqrt{s}$, which thus shows that the $1/\sqrt{s}$ -ratio is tight.

(ii) **Selection Algorithm 7 for SPCA (13).**

Suppose that the output of selection Algorithm 7 is (\hat{S}_1, \hat{S}_2) , similarly, we have

$$w^{spca} = w^* \leq \frac{1}{\sqrt{k \min\{s_1, s_2\}}} \|A_{\hat{S}_1, \hat{S}_2}\|_{(k)} \leq \frac{1}{\sqrt{ks}} \max \left\{ \|A_{\hat{S}_1, \hat{S}_1}\|_{(k)}, \|A_{\hat{S}_2, \hat{S}_2}\|_{(k)} \right\}.$$

The ratio is tight since the following example can achieve it.

Example 7 Suppose that $k = s$, $n = 2s$, and matrix $A \in \mathcal{S}_+^n$ is defined by

$$A = e_1 e_1^\top + \sum_{i \in [s+1, n]} e_i e_i^\top,$$

where for each $i \in [n]$, let $e_i \in \mathbb{R}^n$ denote the i th column of the identity matrix I_n .

In Example 7, the selection Algorithm 7 proceeds as follows: (a) for each $j \in [s]$, we can choose $\hat{S}_1^j = \hat{S}_2^j = [s]$ and the resulting objective value is equal to 1; (b) for each $j \in [s+1, n]$, we can choose $\hat{S}_1^j = \hat{S}_2^j = [2, s] \cup \{j\}$ and the resulting objective value is also equal to 1. In this way, the final objective value of SPCA (13) is equal to 1. However, the true optimal value of SPCA (13) is $\|A_{[s+1, n], [s+1, n]}\|_{(k)} = k = s$, which thus shows that the $1/\sqrt{ks}$ -ratio is tight when $s \leq n/2$.

(iii) **Selection Algorithm 8 for SPCA (13).**

Suppose that the output of selection Algorithm 8 is (\hat{S}_1, \hat{S}_2) , then we have

$$w^{spca} = w^* \leq \frac{\sqrt{s_1 s_2}}{k \sqrt{mn}} \|A_{\hat{S}_1, \hat{S}_2}\|_{(k)} \leq \frac{s}{kn} \max \left\{ \|A_{\hat{S}_1, \hat{S}_1}\|_{(k)}, \|A_{\hat{S}_2, \hat{S}_2}\|_{(k)} \right\}.$$

The ratio is tight since the following example can achieve it.

Example 8 For any positive integer $k > 0, t > 0$, suppose that $s = 2^t$ satisfying $s \geq k + 1$. Let us define $n = 2s$ and matrix $\mathbf{A} \in \mathcal{S}_+^n$ as

$$\mathbf{A} = \sum_{i \in [k]} \frac{n}{s} \begin{pmatrix} \mathbf{0}_s \\ \mathbf{H}_{:,i} \end{pmatrix} \begin{pmatrix} \mathbf{0}_s \\ \mathbf{H}_{:,i} \end{pmatrix}^\top + \begin{pmatrix} \mathbf{1}_s \\ \mathbf{H}_{:,k+1} \end{pmatrix} \begin{pmatrix} \mathbf{1}_s \\ \mathbf{H}_{:,k+1} \end{pmatrix}^\top,$$

where $\mathbf{H} \in \mathbb{R}^{s,s}$ is a Hadamard matrix of order 2^t .

In Example 8, in the construction of matrix \mathbf{A} , the $(k+1)$ vectors are orthogonal because the columns of a Hadamard matrix are orthogonal. It follows that all the non-zero eigenvalues of \mathbf{A} are equal to n since $n = 2s$. Thus, a top eigenvector can be

$$\mathbf{u}_1 = \frac{1}{\sqrt{n}} \begin{pmatrix} \mathbf{1}_s \\ \mathbf{H}_{:,k+1} \end{pmatrix}.$$

Given vector \mathbf{u}_1 , Algorithm 8 selects $\hat{S}_1 = [s]$ at Step 3 and $\hat{S}_2 = [s]$ at Step 4, respectively. Since $\hat{S}_1 = \hat{S}_2 = [s]$, the objective value of SPCA (13) is equal to $\|\mathbf{A}_{[s],[s]}\|_{(k)} = \|\mathbf{1}_{s,s}\|_{(k)} = s$. However, the true optimal submatrix of SSVD (4) is $\mathbf{A}_{[s+1,2s],[s+1,2s]}$ and the resulting optimal value is equal to

$$\|\mathbf{A}_{[s+1,2s],[s+1,2s]}\|_{(k)} = \left\| \sum_{i \in [k]} \frac{n}{s} \mathbf{H}_{:,i} \mathbf{H}_{:,i}^\top + \mathbf{H}_{:,k+1} \mathbf{H}_{:,k+1}^\top \right\|_{(k)} = kn,$$

which confirms that the $s/(kn)$ -approximation ratio is tight. \square

A.4 Proof of Theorem 7

Theorem 7 *The SPCA (16) is equivalent to*

$$(\text{SPCA}) \quad w^{spca} := \max_{\mathbf{z} \in \{0,1\}^n, w \in \mathbb{R}} \left\{ w : \sum_{i \in [n]} z_i \leq s, w \leq \|\mathbf{A}_{S,S}\|_{(k)} + \sum_{i \in [n] \setminus S} A_{ii} z_i, \forall S \subseteq [n], |S| \leq s \right\}. \quad (17)$$

Proof: By separating the binary variables \mathbf{z} , we can rewrite the SPCA (16) as

$$\max_{\substack{\mathbf{z} \in \{0,1\}^n, \\ \sum_{i \in [n]} z_i \leq s}} f(\mathbf{z}) := \max_{\substack{\mathbf{X} \in \mathcal{S}_+^r, \mathbf{X} \preceq \mathbf{I}_r, \\ \mathbf{W}_1, \dots, \mathbf{W}_n \in \mathcal{S}_+^r}} \left\{ \sum_{i \in [n]} \mathbf{c}_i^\top \mathbf{W}_i \mathbf{c}_i : \text{tr}(\mathbf{X}) \leq k, \mathbf{X} \succeq \mathbf{W}_i, \text{tr}(\mathbf{W}_i) \leq z_i, \forall i \in [n] \right\}. \quad (23a)$$

For the inner maximization problem above, we introduce Lagrangian multipliers $\{\mathbf{Q}_i \in \mathcal{S}_+^r\}_{i \in [n]}$ and $\boldsymbol{\mu} \in \mathbb{R}_+^n$ for the constraints with respect to matrix variables $\{\mathbf{W}_i\}_{i \in [n]}$ and the resulting dual problem is

$$f(\mathbf{z}) \leq \min_{\substack{\boldsymbol{\mu} \in \mathbb{R}_+^n, \\ \mathbf{Q}_1, \dots, \mathbf{Q}_n \in \mathcal{S}_+^r}} \max_{\substack{\mathbf{W}_1, \dots, \mathbf{W}_n \in \mathcal{S}_+^r, \\ \mathbf{X} \in \mathcal{S}_+^r, \mathbf{X} \preceq \mathbf{I}_r, \text{tr}(\mathbf{X}) \leq k}} \left\{ \sum_{i \in [n]} \mathbf{c}_i^\top \mathbf{W}_i \mathbf{c}_i + \sum_{i \in [n]} \text{tr}(\mathbf{Q}_i (\mathbf{X} - \mathbf{W}_i)) + \sum_{i \in [n]} \mu_i (z_i - \text{tr}(\mathbf{W}_i)) \right\},$$

where the inequality is by weak duality. Maximizing the inner problem above over variables $\{\mathbf{W}_i\}_{i \in [n]}$ yields

$$f(\mathbf{z}) \leq \min_{\substack{\boldsymbol{\mu} \in \mathbb{R}_+^n, \\ \mathbf{Q}_1, \dots, \mathbf{Q}_n \in \mathcal{S}_+^r}} \max_{\substack{\mathbf{X} \in \mathcal{S}_+^r, \mathbf{X} \preceq \mathbf{I}_r, \text{tr}(\mathbf{X}) \leq k}} \left\{ \sum_{i \in [n]} \text{tr}(\mathbf{Q}_i \mathbf{X}) + \sum_{i \in [n]} \mu_i z_i : \mathbf{c}_i \mathbf{c}_i^\top \preceq \mathbf{Q}_i + \mu_i \mathbf{I}_r, \forall i \in [n] \right\}.$$

Next, from Lemma 2, maximizing the inner problem above over the matrix variable \mathbf{X} yields

$$f(\mathbf{z}) \leq \min_{\mu \in \mathbb{R}_+^n, \mathbf{Q}_1, \dots, \mathbf{Q}_n \in \mathcal{S}_+^r} \left\{ \left\| \sum_{i \in [n]} \mathbf{Q}_i \right\|_{(k)} + \sum_{i \in [n]} \mu_i z_i : \mathbf{c}_i \mathbf{c}_i^\top \preceq \mathbf{Q}_i + \mu_i \mathbf{I}_r, \forall i \in [n] \right\}. \quad (23b)$$

According to the proof of Corollary 3, for any feasible solution \mathbf{z} of SPCA (23a) with its support \hat{S} , the inner maximization problem in (23a) admits the optimal value $f(\mathbf{z}) = \left\| \sum_{i \in \hat{S}} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)}$, implying that the optimal value of its corresponding dual problem (23b) must be lower bounded by $\left\| \sum_{i \in \hat{S}} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)}$ based on the weak duality.

In particular, for a subset $S \subseteq [n]$ of size at most s , we can construct a feasible solution to the dual problem (23b) as

$$\mathbf{Q}_i = \mathbf{c}_i \mathbf{c}_i^\top, \mu_i = 0, \forall i \in S, \mathbf{Q}_i = \mathbf{0}_{r,r}, \mu_i = \mathbf{c}_i^\top \mathbf{c}_i, \forall i \in [n] \setminus S.$$

Plugging the above solution into problem (23b) and finding the minimum of objective values led by them, problem (23b) can be upper bounded by

$$f(\mathbf{z}) \leq \min_{S \subseteq [n], |S| \leq s} \left\{ \left\| \sum_{i \in S} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)} + \sum_{i \in [n] \setminus S} \|\mathbf{c}_i\|_2^2 z_i \right\} \leq \left\| \sum_{i \in \hat{S}} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)} := f(\mathbf{z}), \quad (23c)$$

where the equality can be achieved by the support \hat{S} of \mathbf{z} since

$$\left\| \sum_{i \in \hat{S}} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)} + \sum_{i \in [n] \setminus \hat{S}} \|\mathbf{c}_i\|_2^2 z_i = \left\| \sum_{i \in \hat{S}} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)}.$$

Therefore, SPCA (23a) can be formulated by

$$\max_{\substack{\mathbf{z} \in \{0,1\}^n, \\ \sum_{i \in [n]} z_i \leq s}} \min_{S \subseteq [n], |S| \leq s} \left\{ \left\| \sum_{i \in S} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)} + \sum_{i \in [n] \setminus S} \|\mathbf{c}_i\|_2^2 z_i \right\}. \quad (23d)$$

Introducing the variable w to linearize the inner minimization above and plugging the identities that $\|\mathbf{A}_{S,S}\|_{(k)} = \left\| \sum_{i \in S} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)}$ for any subset $S \subseteq [n]$ and $A_{ii} = \mathbf{c}_i^\top \mathbf{c}_i$ for any $i \in [n]$, we obtain the formulation (17). \square

A.5 Proof of Corollary 6

Corollary 6 For a pair of solutions $(\hat{w}, \hat{\mathbf{z}})$ of SPCA (17), suppose that \hat{S} denotes the support of $\hat{\mathbf{z}}$, then we have

$$\max_{S \subseteq [n], |S| \leq s} \left\{ \hat{w} - \|\mathbf{A}_{S,S}\|_{(k)} - \sum_{i \in [n] \setminus S} A_{ii} \hat{z}_i \right\} = \hat{w} - \|\mathbf{A}_{\hat{S}, \hat{S}}\|_{(k)} - \sum_{i \in [n] \setminus \hat{S}} A_{ii} \hat{z}_i.$$

Proof: First, given the solution $\hat{\mathbf{z}} \in \{0,1\}^n$, we have that

$$\min_{S \subseteq [n], |S| \leq s} \left\{ \|\mathbf{A}_{S,S}\|_{(k)} + \sum_{i \in [n] \setminus S} A_{ii} \hat{z}_i \right\} = \min_{S \subseteq [n], |S| \leq s} \left\{ \left\| \sum_{i \in S} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)} + \sum_{i \in [n] \setminus S} \|\mathbf{c}_i\|_2^2 z_i \right\}, \quad (24)$$

where the minimization problem on right-hand side above, according to the results in (23c), achieves the optimal value

$$\left\| \sum_{i \in \hat{S}} \mathbf{c}_i \mathbf{c}_i^\top \right\|_{(k)} = \|\mathbf{A}_{\hat{S}, \hat{S}}\|_{(k)} = \|\mathbf{A}_{\hat{S}, \hat{S}}\|_{(k)} + \sum_{i \in [n] \setminus \hat{S}} A_{ii} \hat{z}_i,$$

where the last equation is due to $\hat{z}_i = 0$ for all $i \in [n] \setminus \hat{S}$. We thus complete the proof. \square

A.6 Proof of Proposition 4

Proposition 4 *The SSVD (4) is equivalent to*

$$(\text{SSVD}) \quad w^* := \max_{S \subseteq [m+n]} \{ \|\bar{\mathbf{A}}_{S,S}\|_{(k)} : |S \cap [m]| \leq s_1, |S \cap [m+1, m+n]| \leq s_2 \} - k\sigma_1(\mathbf{A}), \quad (18)$$

where matrix $\bar{\mathbf{A}} \in \mathcal{S}_+^{m+n}$ is defined by

$$\bar{\mathbf{A}} \in \mathcal{S}_+^{m+n} := \begin{pmatrix} \mathbf{0}_{m,m} & \mathbf{A} \\ \mathbf{A}^\top & \mathbf{0}_{n,n} \end{pmatrix} + \sigma_1(\mathbf{A})\mathbf{I}_{m+n}.$$

Proof: We show the equivalence of SSVD (4) and problem (18) via one-to-one solution correspondence. First, for any feasible solution (S_1, S_2) to SSVD, let us construct set $S \subseteq [m+n]$ as

$$S := \{i : i \in S_1\} \cup \{j+m : j \in S_2\}.$$

Then we have

$$\begin{aligned} \|\bar{\mathbf{A}}_{S,S}\|_{(k)} &= \left\| \begin{pmatrix} \mathbf{0} & \mathbf{A}_{S_1,S_2} \\ \mathbf{A}_{S_1,S_2}^\top & \mathbf{0} \end{pmatrix} + \sigma_1(\mathbf{A})(\mathbf{I}_{m+n})_{S,S} \right\|_{(k)} = \sum_{i \in [k]} (\sigma_i(\mathbf{A}_{S_1,S_2}) + \sigma_1(\mathbf{A})) \\ &= \|\mathbf{A}_{S_1,S_2}\|_{(k)} + k\sigma_1(\mathbf{A}), \end{aligned} \quad (25)$$

where the first equality is from the definition of matrix $\bar{\mathbf{A}}$ and the second equality is due to Lemma 4.

Second, for any feasible subset S to problem (18), we construct two subsets $S_1 \subseteq [m]$ and $S_2 \subseteq [n]$ as

$$S_1 = \{i : i \in [m] \cap S\}, S_2 = \{j-m : j \in [m+1, m+n] \cap S\},$$

which is feasible to SSVD (4). Similarly, according to (25), we have $\|\mathbf{A}_{S_1,S_2}\|_{(k)} = \|\bar{\mathbf{A}}_{S,S}\|_{(k)} - k\sigma_1(\mathbf{A})$.

This proves the equivalence of SSVD (4) and problem (18). \square