Sample and Expand: Discovering Low-rank Submatrices With Quality Guarantees

Martino Ciaperoni[1⋆](✉), Aristides Gionis[2], and Heikki Mannila[3]

[1] Scuola Normale Superiore, Italy
martino.ciaperoni@sns.it
[2] KTH Royal Institute of Technology, Sweden
argioni@kth.se
[3] Aalto University, Finland
heikki.mannila@aalto.fi

Abstract. The problem of approximating a matrix by a low-rank one has been extensively studied. This problem assumes, however, that the whole matrix has a low-rank structure. This assumption is often false for real-world matrices. We consider the problem of discovering submatrices from the given matrix with bounded deviations from their low-rank approximations. We introduce an effective two-phase method for this task: first, we use sampling to discover small nearly low-rank submatrices, and then they are expanded while preserving proximity to a low-rank approximation. An extensive experimental evaluation confirms that the method we introduce compares favorably to existing approaches.

Keywords: Low-rank approximation · submatrix detection.

1 Introduction

Low-rank approximation has emerged as a fundamental task in many data-analysis applications, including machine-learning pipelines [26], large language models [14], recommender systems [16], image compression and denoising [12]. The goal of low-rank approximation is to represent an input matrix as accurately as possible using a small number of row and column vectors.

For decades, the *singular value decomposition* (SVD), with the closely related principal component analysis (PCA), has remained the gold standard for low-rank approximation [11]. Despite its success, SVD has certain limitations. Among others, when applying SVD we aim to find a low-rank approximation for the entire input matrix. This assumption can be rather restrictive, as in real-world data it might be that only certain submatrices are well approximated by low-rank structures. For instance, in ratings data originating in movie recommender systems, low-rank submatrices occur because subsets of users may share a similar taste only for a subset of movies. Similar local patterns could be observed in data coming from other domains, such as market-basket analysis, image processing, and biology [6]. The SVD can fail to identify local low-rank submatrices.

⋆ The work was done while the author was at Aalto University.

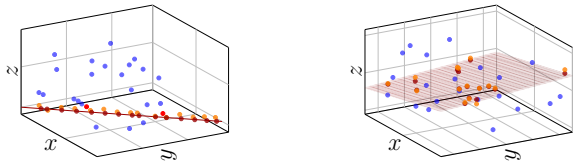

Fig. 1: Example. A subset of data points (in orange) in the 3-dimensional space are close to their projection (in red) onto a line in the xy-plane (a) or to a plane in the 3-dimensional space (b), while other points (in blue) can be further away.

Existing approaches to find local low-rank submatrices. The task of identifying submatrices that are well described by low-rank structures has been largely overlooked until recently [6]. Existing work in this direction is based primarily on the SVD, and does not provide any guarantee on the quality of the approximation for the identified submatrices.

Our approach. In this work, we adopt a different perspective on discovering local low-rank patterns, and we address the problem of identifying submatrices that are guaranteed to be close to a low-rank approximation. Our quality guarantees hold with respect to an approximation that can be easily interpreted in terms of the original data, which can be particularly valuable for applications in different domains. For example, near-rank-1 submatrices can be accurately approximated by each row in the submatrix being colinear with a single row. Unlike previous work, our work does not directly rely on the SVD. Nearly-low-rank submatrices correspond to subsets of points (matrix rows) that approximately lie on a low-dimensional subspace, for a subset of dimensions (matrix columns). For rank equal to 1, which is a particularly interesting case, the points approximately lie on a line through the origin, and for rank equal to 2 the points are close to a plane through the origin, as in the example of Figure 1, which shows data points identifying a 15×2 near-rank-1 submatrix and a 15×3 near-rank-2 submatrix.

While SVD may fail to reveal dense lines in the data, it is possible to find such structures by sampling. A naïve approach would be to sample subsets of points and dimensions until a large set of nearly-collinear points is found. However, this procedure quickly becomes inefficient. Instead, to identify points approximately distributed along a line, we introduce a method that only relies on sampling in an initialization phase to find a minimal structure that can exhibit this property, i.e., two points in two-dimensions. In a subsequent phase, the 2×2 submatrix is expanded deterministically to obtain the entire subset of points and dimensions associated with the target line. Based on such a two-phase method, we discover arbitrary submatrices that admit low-rank approximations that can be easily interpreted in terms of the original submatrix rows or columns, and are supported by quality guarantees. A real-world example is given in Figure 2.

Our contributions. Our main contributions can be summarized as follows.

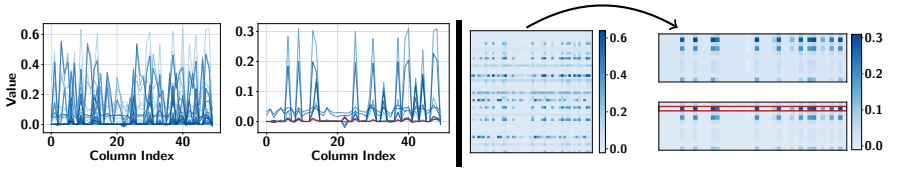

Fig. 2: HYPERSPECTRAL dataset. On the left, we show the values of the rows in a 50×50 matrix and the nearly-proportional values of the rows in a near-rank-1 11×44 submatrix discovered by our method. On the right we show the matrix and, next to it, the discovered submatrix (top) and its accurate approximation expressing each row as collinear with the row highlighted in red (bottom).

- We formalize the problem of finding submatrices that are provably close to a low-rank approximation.
- We introduce an effective method for finding submatrices that are provably close to rank 1. Then, we generalize this method to the rank-k case.
- We analyze the theoretical properties of the method we introduce.
- We demonstrate the advantages of our method over previous work.

Roadmap. The rest of the paper is organized as follows. In Section 2 we discuss related work. Section 3 introduces the notation used throughout the paper as well as key preliminary concepts. In Section 4 we present the problem we study and in Section 5 we illustrate our method to address it. In Section 6 we analyze the properties of the method, and in Section 7 we assess its empirical performance. Finally, Section 8 provides conclusions.

2 Related Work

Low-rank matrix approximation. Low-rank approximation techniques are widely used to decompose a matrix into simpler components, capturing essential patterns while reducing noise and dimensionality. The SVD and the related PCA are among the most popular techniques [11]. Nonnegative matrix factorization techniques [9] have become popular in applications where the goal is to decompose data in nonnegative components. Boolean matrix decomposition relies on boolean algebra instead of linear algebra [21]. Column subset selection [3] and the CUR decomposition [19] have emerged as more interpretable alternatives to the SVD. In 2019, Gillis and Shitov [10] studied the problem of low-rank approximation to minimize the maximum entry-wise deviation. More recently, an approach to low-rank approximation that accounts for multiplicative effects was introduced [4].

Local low-rank matrix approximation. Relatively less research has been conducted for finding decompositions that do not assume a *global* low-rank structure, which is the focus of our paper. The goal here is to find submatrices that are *locally* well-approximated by a low-rank structure. A simple heuristic to local low-rank approximation is obtained by imposing a sparsity constraint on matrix decomposition, and sparse PCA [20] is a prominent example of such methods.

Doan and Vavasis proposed the problem of recovering near-rank-1 submatrices by framing it as a convex-optimization problem [8]. Lee et al. [16] introduced the LLORMA method to address matrix-completion tasks while relaxing the assumption that the entire matrix has low rank. LLORMA approximates the entire input matrix, and thus, it is fundamentally different from our work, which focuses on detecting local low-rank patterns. On the other hand, the problem we study finds application in matrix completion, as shown by the work of Ruchansky et al. [22], which introduces the SVP method to quickly detect low-rank submatrices with the ultimate goal of improving the accuracy in matrix completion. While SVP cannot discover arbitrary low-rank submatrices, Dang et al. [6] introduced the RPSP method, which addresses this lack of generality. The core idea behind RPSP is to sample submatrices and count the number of times that each entry belongs to a low-rank submatrix. Like RPSP, our method targets arbitrary near-low-rank submatrices. Unlike previous methods, our method can in principle identify submatrices that are close to a specific target rank.

Co-clustering, projective clustering, and subspace clustering. Co-clustering algorithms [7] simultaneously cluster the rows and columns of a matrix. Although co-clustering algorithms can be used for detecting low-rank submatrices, they cannot generally discover such structures, except in specific cases where the values of the low-rank submatrices deviate significantly from the background. Projective clustering and subspace clustering are also related problems. In projective clustering [1], the goal is to partition the data into subsets such that the points in each subset are close to each other in some subspace. In subspace clustering [25] the goal is to find a representation of the input data as a union of different subspaces. In general, clustering problems are fundamentally different from the problem we study, as they seek a partitioning of the entire data matrix.

3 Preliminaries

Notation and basic definitions. Matrices are denoted by upper-case boldface letters, and we use $\mathbf{D}$ to denote the input data matrix. $\mathbf{D}_{i,j}$ indicates the entry of $\mathbf{D}$ in row i and column j, while the i-th row and j-th column of $\mathbf{D}$ are denoted by $\mathbf{D}_{i,:}$ and $\mathbf{D}_{:,j}$, respectively. Sets are denoted by upper-case letters and scalars by lower-case letters. Vectors are denoted by lower-case boldface letters, e.g., $\mathbf{x} = (x_1, \ldots, x_d)$. We denote the L_2 (or Euclidean) norm of a vector $\mathbf{x}$ as $\|\mathbf{x}\|_2$ and the inner product between two vectors $\mathbf{x}$ and $\mathbf{y}$ as $\mathbf{x}^T\mathbf{y}$. We consider different matrix norms: the Frobenius norm $\|\mathbf{D}\|_F = (\sum_{i,j} |\mathbf{D}_{i,j}|^2)^{1/2}$, the spectral norm $\|\mathbf{D}\|_2 = \sup_{||\mathbf{x}||_2 \leq 1} ||\mathbf{D}\mathbf{x}||_2$, and the max norm $\|\mathbf{D}\|_{\max} = \max_{i,j} |\mathbf{D}_{i,j}|$. We use $\|\mathbf{D}\|_*$ with $* = \{2, F, \max\}$ to indicate the above norms. We refer to the total number of entries of a (sub)matrix $\mathbf{D}$ as its *size*, which is also simply denoted by $|\mathbf{D}|$, if there is no risk of confusion with the entry-wise absolute value. For a matrix $\mathbf{X}$, we denote by $\hat{\mathbf{X}}$ a low-rank approximation of $\mathbf{X}$ and by $\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}$ the difference $\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}} = \mathbf{X} - \hat{\mathbf{X}}$.

Orthogonal projections. Given a nonzero vector $\mathbf{x}$ and a vector $\mathbf{y}$, the *orthogonal projection* of $\mathbf{y}$ onto $\mathbf{x}$ is given by $\text{Proj}_{\mathbf{x}}\,\mathbf{y} = (\mathbf{y}^T\mathbf{x})/(\mathbf{x}^T\mathbf{x})\mathbf{x}$. Similarly, given

a matrix $\mathbf{B}$ and a matrix $\mathbf{A}$ with linearly independent columns, the orthogonal projection of $\mathbf{B}$ onto the column space of $\mathbf{A}$ is given by $\text{Proj}_{\mathbf{A}}\,\mathbf{B} = \mathbf{A}\mathbf{A}^{+}\mathbf{B}$, where $\mathbf{A}^{+} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the *Moore-Penrose pseudoinverse* of $\mathbf{A}$.

Low-rank approximation and SVD. The *singular value decomposition* (SVD) of a matrix $\mathbf{D} \in \mathbb{R}^{n\times m}$ is given by $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{m\times m}$ are unitary matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{n\times m}$ is a diagonal matrix with singular values $\{\sigma_1, \sigma_2 \ldots \sigma_{\min\{n,m\}}\}$ as diagonal entries, conventionally sorted in decreasing order. If the matrix is not clear from the context, we denote as $\sigma_i(\mathbf{X})$ the i-th singular value of $\mathbf{X}$. It is known that the optimal rank-k approximation of $\mathbf{D}$ for the Frobenius and the spectral norm (but not for the max norm) is obtained from the SVD by retaining the first k singular values, along with the associated k columns of $\mathbf{U}$ and rows of $\mathbf{V}^T$. The largest singular value of a matrix equals its spectral norm, and the number of non-zero singular values indicates the rank.

As real-world data are often noisy, the singular values are seldom exactly zero. Accordingly, to measure the proximity of a matrix to rank 1, in this work, we use the *low-rankness score* [6], which is given by $\ell r(\mathbf{X}) = \frac{\sigma_1(\mathbf{X})^2}{\sum_{i=1}^{\min(n,m)} \sigma_i(\mathbf{X})^2}$. A matrix whose singular values after the k-th one are close to zero can be accurately approximated by a rank-k matrix, and is loosely referred to as *near-rank-k* matrix.

4 Problem Formulation

Next, we formalize the problems we study in this paper. To provide better insight, we first present a special case, and then introduce the more general problems.

Searching for a near-rank-1 subset of rows or columns. As a warm-up, we first introduce a simple problem that fixes the matrix columns or rows.

*Problem 1 (Largest near-rank-*1 *subset of rows (LNROSR)).* Given a matrix $\mathbf{D} \in \mathbb{R}^{n\times m}$ with set of rows $\mathcal{R}$ and a threshold $\epsilon \in \mathbb{R}^{+}$, find the largest subset of rows $\mathcal{R}' \subseteq \mathcal{R}$ such that there exist a rank-1 matrix $\hat{\mathbf{X}} \propto \mathbf{x}\mathbf{y}^T$, where $\mathbf{y}^T \in \mathbb{R}^m$ is a row of $\mathbf{D}$, satisfying

$$\|\mathbf{D}_{i,:} - \hat{\mathbf{X}}_{i,:}\|_2 \leq \epsilon, \qquad \text{for all } \; i \in \mathcal{R}'. \tag{1}$$

Problem 1 asks to find the largest near-rank-1 submatrix defined over a subset of rows of $\mathbf{D}$ and all columns. This problem is computationally tractable.

Proposition 1. *The LNROSR problem can be solved in polynomial time.*

The proof, via a simple algorithm, is presented in Appendix A of the extended version of the paper [5].

While Problem 1 asks for a subset of rows, the symmetric problem asking for a subset of columns can be solved by considering $\mathbf{D}^T$ in place of $\mathbf{D}$.

Searching for a near-rank-1 submatrix. Next, we discuss the more challenging problem of finding a near-rank-1 submatrix, without fixing neither the rows nor the columns of the input matrix.

Problem 2 (Largest near-rank-1 submatrix (LNROS)). Given a matrix $\mathbf{D} \in \mathbb{R}^{n \times m}$ and a threshold $\epsilon \in \mathbb{R}^+$, find a submatrix $\mathbf{X} \in \mathbb{R}^{n' \times m'}$ of maximum size such that there exist a rank-1 matrix $\hat{\mathbf{X}}$ satisfying

$$\|\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}\|_* = \|\mathbf{X} - \hat{\mathbf{X}}\|_* \leq \epsilon, \quad \text{where } * \text{ can be any of the norms } \{F, 2, \max\}. \tag{2}$$

Unfortunately, due to the interaction between rows and columns, the LNROS problem is computationally intractable.

Proposition 2. *The LNROS problem is* **NP***-hard.*

The **NP**-hardness of LNROS follows from that of the largest rank-1 submatrix problem [8] by setting $\epsilon = 0$, and highlights the connection with the *maximum-edge biclique* problem [18], which is made evident in Section 5.

Searching for a near-rank-k submatrix. We generalize the LNROS problem to the case of near-rank-k submatrices.

Problem 3 (Largest near-rank-k submatrix (LNRkS)). Given a matrix $\mathbf{D} \in \mathbb{R}^{n \times m}$ and a threshold $\epsilon \in \mathbb{R}^+$, find a submatrix $\mathbf{X} \in \mathbb{R}^{n' \times m'}$ of maximum size such that there exist a rank-k matrix $\hat{\mathbf{X}}$ satisfying

$$\|\mathbf{X} - \hat{\mathbf{X}}\|_* \leq \epsilon, \quad \text{where } * \text{ can be any of the norms } \{F, 2, \max\}. \tag{3}$$

As LNRkS is a generalization of LNROS, the LNRkS problem is also **NP**-hard.

Extensions. The problem formulations presented above focus on extracting a single submatrix. In practice, one may wish to find a representation of the input matrix as a sum of N local low-rank patterns. Such a problem is a generalization of both LNROS and LNRkS, and hence, inherits their hardness.

Additionally, it may be of interest to identify submatrices that define affine subspaces. Extending our problem formulations and method to the case of affine subspaces (or near-low-rank submatrices up to a particular translation) is straightforward. The details are deferred to an extended version of this work.

5 Algorithms

In this section, we present SampleAndExpand, our method for discovering near-low-rank submatrices. We first give an overview of the method, and then we present the algorithms to detect near-rank-1 and near-rank-k submatrices.

5.1 High-level Overview of the Method

SampleAndExpand is based on a simple two-phase procedure. The first phase *samples* small seed submatrices, and the second phase *expands* those seed submatrices into larger near-low-rank submatrices.

The main idea relies on the simple principle that any submatrix of a rank-k matrix must also have rank at most k. Thereby, a near-rank-1 submatrix $\mathbf{X}$ of

Algorithm 1 Overview of SampleAndExpand.

Input: Matrix $\mathbf{D}$, target rank k, number of initializations N_{init}, initial tolerance δ_{init}, tolerance δ.
Output: Near-rank-k submatrix $\mathbf{X}^*$.
$\mathbf{X}^* \leftarrow \mathbf{0}$
for $i = 1$ to N_{init} **do**
  $\mathbf{P} \leftarrow$ **Initialization**($\mathbf{D}$, k, δ_{init}) *// first phase: initialization (sampling)*
  $\mathbf{X} \leftarrow$ **Expansion**($\mathbf{P}$, k, δ) *// second phase: expansion*
  if $f(\mathbf{X}) \geq f(\mathbf{X}^*)$ **then**
    $\mathbf{X}^* \leftarrow \mathbf{X}$ *// select the best submatrix across different initializations*
  end if
end for
Return $\mathbf{X}^*$

size $n' \times m'$ contains a large number of 2×2 near-rank-1 submatrices. Thus, if we are looking for a rank-1 submatrix, in the first phase (initialization phase) we identify a *seed*, which is a 2×2 submatrix that can be expanded into a larger near-rank-1 submatrix. The goal of the second phase (expansion phase) is to expand the seed into a large near-rank-1 submatrix. Similarly, if we are looking for a near-rank-k submatrix, in the first phase we identify a seed submatrix of minimal size that is close to rank k, and then we expand it as much as possible.

The two-phase procedure is repeated N_{init} times, to explore different random initializations. Each repetition outputs a nearly low-rank submatrix $\mathbf{X}$. SampleAndExpand accepts a parameter δ that controls the trade-off between proximity to the target rank and size of the output matrices. Higher values of δ tend to yield submatrices that are larger but deviate more from their low-rank approximation.

After the last repetition, we return the submatrix $\mathbf{X}$ that maximizes the objective function $f(\mathbf{X}) = |\mathbf{X}| - \frac{\lambda}{|\mathbf{X}|}\|\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}\|_F^2$. By default, in the absence of prior information, we standardize the error term and the size, and set $\lambda = 1$. However, SampleAndExpand is flexible and supports any objective function.

The pseudocode of the high-level SampleAndExpand method is given in Algorithm 1. The details of the sampling and expansion phase of the method for the rank-1 case and for the general rank-k case are described later.

Approximating the discovered submatrices. The discovered submatrices can be approximated via SVD. Further, SampleAndExpand also naturally leads to a low-rank approximation that is more interpretable than the SVD since it is based on the rows (or columns) of $\mathbf{X}$. For the rank-1 case, this approximation is given by $\hat{\mathbf{X}} = \mathbf{x}\mathbf{y}^T$, where, either $\mathbf{y}^T$ is a row or $\mathbf{x}$ a column of $\mathbf{X}$. If, e.g., $\mathbf{y}^T$ is a row of $\mathbf{X}$, $\mathbf{x}$ can be chosen to minimize $\|\mathbf{X} - \mathbf{x}\mathbf{y}^T\|_F^2$. The resulting optimal $\mathbf{x}$ is the vector of coefficients that describe the orthogonal projections of the rows of $\mathbf{X}$ onto $\mathbf{y}^T$. An analogous argument also applies to the columns. As discussed in Section 6, this approximation is supported by approximation guarantees.

Although the rank-1 SVD may be more accurate than the interpretable alternative, if a matrix is sufficiently close to rank 1, the difference is often negligible. To gain some intuition for this claim, note that a matrix that has

exactly rank 1 can be represented with no error not only by the discussed interpretable approximation, but also by the rescaled outer product $\alpha \mathbf{X}_{:,j} \mathbf{X}^T_{i,:}$ of any of its rows and columns, for some $\alpha \in \mathbb{R}$. If instead the matrix deviates significantly from rank 1, the rank-k interpretable approximation based on orthogonal projections is often not as accurate as the rank-k SVD.

Discovering multiple submatrices. In practice, we may wish to discover multiple submatrices within a single matrix $\mathbf{D}$ and eventually obtain an approximation $\hat{\mathbf{D}}$ of the matrix as sum of local low-rank patterns. To achieve this, we run SampleAndExpand iteratively. In each iteration, the method finds a single near-rank-k submatrix, and then updates $\hat{\mathbf{D}}$ and the input matrix. This simple procedure is summarized in Algorithm 2 of the extended version of the paper [5].

5.2 Recovering a Near-rank-1 Submatrix

Here, we present the initialization (sampling) and expansion phases of the algorithm to discover near-rank-1 submatrices. Algorithm 3 in the extended version of the paper [5] presents the pseudocode.

Initialization. To find the initial 2×2 near-rank-1 submatrix $\mathbf{P}$, we sample two distinct row indices $\{i_1, i_2\}$ and column indices $\{j_1, j_2\}$ of the input matrix $\mathbf{D}$, and then we compute the determinant of the associated 2×2 submatrix $\mathbf{P}'$:

$$\det(\mathbf{P}') = \mathbf{D}_{i_1,j_1}\mathbf{D}_{i_2,j_2} - \mathbf{D}_{i_1,j_2}\mathbf{D}_{i_2,j_1}.$$

If $|\det(\mathbf{P}')| \leq \delta_{init}$, for some input $\delta_{init} \in \mathbb{R}^+$, $\mathbf{P}'$ is close to rank 1, and hence it may be contained into a larger near-rank-1 submatrix. Therefore, $\mathbf{P}'$ is the seed $\mathbf{P}$ that will be expanded. If instead $|\det(\mathbf{P}')| > \delta_{init}$, we sample different 2×2 submatrices $\mathbf{P}'$ until we find a seed to expand. In practice, δ_{init} is initialized to a small value (10^{-11} by default) and progressively increased until the seed is found.

Expansion. To extend $\mathbf{P}$ into a larger submatrix, we consider one of the entries (i_a, j_a) in $\mathbf{P}$, which we call *anchor*. Then, we divide all rows in $\mathbf{D}$ by the i_a-th row, obtaining the row-wise ratio matrix $\mathbf{R}^r$ and all columns by the j_a-th column, obtaining the column-wise ratio matrix $\mathbf{R}^c$. If an entry in the i_a-th row or j_a-th column of $\mathbf{D}$ is zero, we add a small positive constant to prevent division by zero.

The ratios indicate which entries may belong to a near-rank-1 submatrix with the anchor. As illustrated in Figure 3, if the matrix $\mathbf{D}$ contains a submatrix $\mathbf{X}$ of rank 1, the entries corresponding to $\mathbf{X}$ in $\mathbf{R}^r$ and $\mathbf{R}^c$ will be row-wise and column-wise constant, respectively. More generally, as we explain in Section 6, bounding the variation across rows of row-wise and columns of column-wise ratios in a submatrix leads to quality guarantees for its rank-1 approximation. Thus, the goal of the expansion stage is to identify a submatrix of maximum size with bounded variation in the rows of the row-wise ratios and in the columns of the column-wise ratios. To this end, our algorithm examines the rows of $\mathbf{R}^r$ to find subsets of near-constant entries including column j_a and the columns of $\mathbf{R}^c$ to find subset of near-constant entries including row i_a. More specifically, for an input parameter $\delta \in \mathbb{R}^+$, we select, for each row of $\mathbf{R}^r$, the j_a-th entry and all

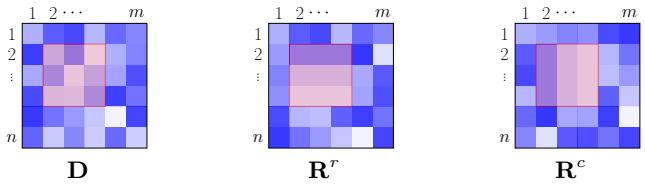

Fig. 3: Example of row-wise ($\mathbf{R}^r$) and column-wise ($\mathbf{R}^c$) ratio matrices associated with an input matrix ($\mathbf{D}$) containing a rank-1 submatrix (highlighted in red). Within this rank-1 submatrix, the entries of $\mathbf{R}^r$ are constant across rows, and the entries of $\mathbf{R}^c$ are constant across columns.

other entries such that the maximum variation is less than δ in absolute value. Similarly, for each column of $\mathbf{R}^c$, we select the i_a-th entry and all other entries such that the maximum variation is less than δ in absolute value. Subsets within each row can be efficiently retrieved by sorting the row elements by their absolute deviation from the j-th element and similarly for the columns.

Given the identified subsets, we construct two indicator matrices: $\mathbf{I}^r \in \{0,1\}^{n\times m}$, where the entries with value 1 correspond to subsets of near-constant row-wise ratios; and $\mathbf{I}^c \in \{0,1\}^{n\times m}$, where the entries with value 1 correspond to subsets of near-constant column-wise ratios. We can then compute the intersection of the two matrices $\mathbf{I}^r$ and $\mathbf{I}^c$ to obtain the intersection indicator matrix $\mathbf{I}$ of the same dimension. The problem of extracting a submatrix of maximum size with all row-wise and column-wise ratios with bounded deviations can then be framed as the problem of finding the largest possible all-ones submatrix within $\mathbf{I}$. This problem is equivalent to the extraction of a maximum-edge biclique [18] from the bipartite graph $\mathcal{G}_{\mathbf{I}}$ that has $\mathbf{I}$ as adjacency matrix. Although this is an **NP**-hard problem [18], so that it cannot be solved in polynomial time, we can leverage recent algorithmic advances that solve the problem quickly in considerably dense and large bipartite graphs [18]. In addition, to avoid possible scalability issues that may still arise, we also rely on effective heuristics, as discussed in Section 5.4.

5.3 Recovering a Near-rank-k Submatrix

Next, we illustrate the adaptation of the initialization and expansion phases of SampleAndExpand to the general case of recovery of near-rank-k submatrices. The pseudocode is given as Algorithm 4 in the extended version of the paper [5].

Initialization. The goal of the initialization phase is to identify a seed, i.e., a minimal near-rank-k submatrix to be expanded at a later time. To find the seed, we sample matrices $\mathbf{P}'$ with $k+1$ rows and columns until we find a seed matrix $\mathbf{P}$ that is close to rank k or lower. In order to determine whether a $k+1 \times k+1$ matrix $\mathbf{P}'$ has rank k or lower, we check whether $|\det(\mathbf{P}')| \leq \delta_{init}$, for a small $\delta_{init} \in \mathbb{R}^+$, which, as for the algorithm tailored to near-rank-1 submatrices, is first initialized to a to a small value, and then increased until the seed is found.

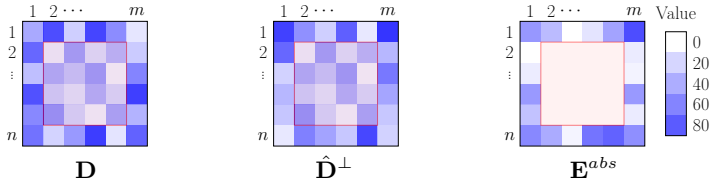

Fig. 4: Example of matrices of orthogonal projections ($\hat{\mathbf{D}}^{\perp}$) and absolute errors ($\mathbf{E}^{abs}$) associated with an input matrix ($\mathbf{D}$) containing a rank-k submatrix (highlighted in red). Within this rank-k submatrix, the entries of $\hat{\mathbf{D}}^{\perp}$ are equal to those of $\mathbf{D}$, and the entries of $\mathbf{E}^{abs}$ are therefore all identically zero.

Expansion. Given the seed matrix $\mathbf{P} \in \mathbb{R}^{k+1 \times k+1}$, of rank $k' \leq k$, we sample k' anchor rows from the rows of $\mathbf{P}$. Let $\mathcal{C}_{\mathbf{P}}$ denote the set of the $k+1$ indices of the columns in $\mathbf{P}$. Considering only the columns in $\mathcal{C}_{\mathbf{P}}$, we compute the coefficients of the orthogonal projection of each row of $\mathbf{D}$ onto the subspace spanned by the k' anchor rows. We then compute the orthogonal projections $\hat{\mathbf{D}}^{\perp} \in \mathbb{R}^{n \times m}$ expressing each row as a linear combination of the anchor rows with weights given by the orthogonal-projection coefficients. The coefficients are obtained by considering only the columns in $\mathcal{C}_{\mathbf{P}}$ identified in the initialization phase. Nevertheless, the matrix $\hat{\mathbf{D}}^{\perp} \in \mathbb{R}^{n \times m}$, similarly to the ratio matrices in the rank-1 case, indicates which additional columns and rows are close to a rank-k approximation. More specifically, all entries $\mathbf{D}_{i,j}$ that are closely approximated by $\hat{\mathbf{D}}^{\perp}_{i,j}$ lie close to the k-dimensional subspace identified in the initialization phase.

Therefore, to find a near-k submatrix of maximum size, which is the goal of the expansion phase, we need to identify the largest submatrix of $\hat{\mathbf{D}}^{\perp}_{i,j}$ where all entries nearly match the corresponding entries of $\mathbf{D}$. To obtain such a submatrix, we calculate the matrix of absolute errors $\mathbf{E}^{abs} = |\mathbf{D} - \hat{\mathbf{D}}^{\perp}|$, and from it, the indicator matrix $\mathbf{I}^r$, which takes value 1 for entry (i,j) if $\mathbf{E}^{abs}_{i,j} \leq \delta$ and 0 otherwise, for some input $\delta \in \mathbb{R}^{+}$. Figure 4 presents an example of matrices $\hat{\mathbf{D}}^{\perp}$ and $\mathbf{E}^{abs}$.

The same procedure followed to determine $\mathbf{I}^r$, but on input $\mathbf{D}^T$ and $\mathbf{P}^T$ yields $\mathbf{I}^c$. The intersection of $\mathbf{I}^r$ and $\mathbf{I}^c$ gives the matrix $\mathbf{I}$ and the associated bipartite graph $\mathcal{G}_{\mathbf{I}}$. As for the case of near-rank-1 submatrix discovery, the desired output near-rank-k submatrix is then given by a submatrix of $\mathbf{I}$ consisting of all ones, or, equivalently, by a maximum-edge biclique of $\mathcal{G}_{\mathbf{I}}$.

5.4 Scalability Considerations

One limitation of SampleAndExpand is its reliance on solving the maximum-edge biclique problem, which is **NP**-hard. While the algorithm we use to extract these bicliques is often efficient in practice [18], scalability issues may still arise. To address such issues, the algorithm for finding maximum-edge bicliques can be replaced with a more scalable heuristic. Among many possible different heuristic approaches, by default, we rely on spectral biclustering [15], which is empirically found to be particularly effective in quickly identifying a dense submatrix of $\mathbf{I}$.

Even more efficient and scalable approaches include algorithms to extract dense bipartite subgraphs, a greedy algorithm removing rows and columns from $\mathbf{I}$, e.g., based on the amount of ones, or a randomized algorithm sampling submatrices from $\mathbf{I}$ according to the amount of ones they contain [2]. A comprehensive evaluation of the performance of various heuristics for approximating maximum-edge bicliques in SampleAndExpand is left to future work.

6 Analysis

In this section, we explain how the proposed methods yield submatrices with bounded approximation error. We also provide a brief discussion on the probabilistic aspects and on the computational complexity of the methods.

6.1 Approximation Error Guarantees

In global low-rank approximation, the presence of outliers in the data may lead to situations where the whole matrix cannot be approximated with a low-rank structure without compromising the overall approximation quality. However, as our problem definition lifts the requirement that the whole matrix must be approximated, it is interesting to control the *entry-wise* maximum approximation error in the discovered submatrices. We thus provide approximation-error guarantees in terms of the max norm. A bound on the max norm yields bounds on the Frobenius and spectral norms, albeit loose. In the case of near-rank-1 submatrix discovery, we also provide interesting bounds on the spectral and Frobenius norms that are not a direct consequence of the bound on the max norm.

Near-rank-1 submatrices. As mentioned in Section 5.1, $n' \times m'$ near-rank-1 submatrices contain many near-rank-1 2×2 submatrices. Building on this intuition, SampleAndExpand starts by locating a 2×2 submatrix $\mathbf{P}$ with bounded determinant, and hence close to rank 1. Then, it computes row-wise and column-wise ratios dividing all rows (columns) by a single anchor row (column) with index sampled from those of $\mathbf{P}$, and finds submatrices with rows (columns) of nearly-constant ratios. Nearly-constant ratios correspond to bounded 2×2 determinants. For instance, if $\left| \frac{\mathbf{D}_{i,j_1}}{x^r_{j_1}} - \frac{\mathbf{D}_{i,j_2}}{x^r_{j_2}} \right| \leq \delta$, then $|\mathbf{D}_{i,j_1} x^r_{j_2} - \mathbf{D}_{i,j_2} x^r_{j_1}| \leq \delta |x^r_{j_1}||x^r_{j_2}|$ where the left-hand side is a 2×2 determinant. Bounding the variation of all ratios within each row and column, and thus the corresponding 2×2 determinants, SampleAndExpand yields submatrices composed of 2×2 near-rank-1 submatrices, which, as formalized in Theorem 1, results in approximation guarantees.

Theorem 1. *Let $\mathbf{X} \in \mathbb{R}^{n' \times m'}$ be a near-rank-1 submatrix output by SampleAndExpand with anchor row $\mathbf{x}^r$, anchor column $\mathbf{x}^c$ and input tolerance δ. There exists a rank-1 approximation $\hat{\mathbf{X}}$ of $\mathbf{X}$ such that for $\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}} = \mathbf{X} - \hat{\mathbf{X}}$ it holds:*

$$\|\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}\|_{\max} \leq \min \left\{ \delta g_{\max}(\mathbf{x}^r), \delta g_{\max}(\mathbf{x}^c) \right\}, \tag{4}$$

and

$$\|\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}\|_2 \leq \|\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}\|_F \leq \min \left\{ \delta \sqrt{(n-1) g_F(\mathbf{x}^r)}, \delta \sqrt{(m-1) g_F(\mathbf{x}^c)} \right\}, \tag{5}$$

where $g_{\max}(\mathbf{x}) = \frac{\max_i |x_i|^3}{2\min_i x_i^2}$ *and* $g_F(\mathbf{x}) = \frac{\sum_{i<j} {x_i}^2 {x_j}^2}{\|\mathbf{x}\|_2^2}$.

Theorem 1 suggests that the low-rank-approximation error incurred by the near-rank-1 submatrices discovered by SampleAndExpand can be bounded by a function of the input parameter δ and of the scale of $\mathbf{x}^r$ or $\mathbf{x}^c$. Therefore, given the anchor row and column, one can set the value of δ to guarantee that the maximum or the total approximation error is bounded by a user-specified threshold $\epsilon \in \mathbb{R}^+$, as requested by Problem 2. However, the approximation-error guarantees given in Theorem 1 only hold if SampleAndExpand extracts a biclique in the last step. Alternative heuristic approaches that do not extract a biclique can be effective in practice, but they are not supported by approximation-error guarantees.

Notably, the approximation-error guarantees are achieved by the interpretable rank-1 approximation discussed in Section 5.1. In addition, for the rank-1 SVD approximation $\hat{\mathbf{X}}$, Theorem 2 in Appendix B of the extended version of the paper [5] bounds the spectral norm of the error $\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}} = \mathbf{X} - \hat{\mathbf{X}}$.

Near-rank-k submatrices. The algorithm for the more general task of identifying near-rank-k submatrices does not admit the same analysis as the algorithm for identifying near-rank-1 submatrices. However, the algorithm for the rank-k case, by design, discovers submatrices $\mathbf{X}$ such that $\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}} = \mathbf{X} - \hat{\mathbf{X}}$ satisfies $\|\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}\|_{\max} \leq \delta$. As mentioned, the bound on the max norm leads to a straightforward bound on the Frobenius and spectral norms, namely $\|\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}\|_2 \leq \|\mathbf{E}_{\mathbf{X},\hat{\mathbf{X}}}\|_F \leq \delta\sqrt{(n-k)(m-k)}$, which can also be used to set the value of δ based on user-specified error threshold ϵ on the Frobenius or spectral norm.

6.2 Probabilistic Analysis

In this section, we discuss simple probabilistic aspects of our method.

Probability of discovering a near-rank-1 submatrix. Let $\mathbf{X}$ be a target near-rank-1 submatrix of size $|\mathbf{X}|$ within $\mathbf{D} \in \mathbb{R}^{n\times m}$. The probability that SampleAndExpand discovers $\mathbf{X}$ by one sample is $p = \frac{|\mathbf{X}|}{nm}\frac{|\mathbf{X}|-1}{nm-1}$. Hence, the probability of discovering $\mathbf{X}$ in N_{init} iterations is $1-(1-p)^{N_{init}}$, and therefore the number of iterations required to discover $\mathbf{X}$ with probability at least α_p is $N_{init} \geq \frac{\ln(1-\alpha_p)}{\ln(1-p)}$. For instance, if $p = 0.1$ and $\alpha_p = 0.9$, we need $N_{init} > \frac{\ln(1-0.9)}{\ln(1-0.1)} \approx 22$ iterations.

Basic probability theory implies that, in expectation, the number of iterations necessary to discover $\mathbf{X}$ is $\frac{1}{p}$, and we discover it pN_{init} times in N_{init} iterations.

Probability of discovering a near-rank-k submatrix. The simple probabilistic analysis presented above for near-rank-1 submatrices also applies to near-rank-k submatrices. The only difference is that, in this case, we have $p = \frac{|\mathbf{X}|}{nm}\frac{|\mathbf{X}|-1}{nm}\ldots\frac{|\mathbf{X}|-k}{nm-k}$, which can become small as k grows. Yet, larger values of k tend to be associated with larger values of $|\mathbf{X}|$ and, in practice, we are interested in small values of k.

Probability of occurrence of a 2×2 near-rank-1 matrix. We conclude the section by investigating the probability with which SampleAndExpand

identifies a seed 2×2 submatrix with near-zero determinant in random matrices. Let $\mathbf{D}$ be a random matrix with i.i.d. entries distributed according to $\mathcal{Z}$, and let $E(\mathcal{Z}) = \mu$ and $Var(\mathcal{Z}) = \sigma^2$ be the expectation and variance of $\mathcal{Z}$. To study the probability of occurrence of 2×2 submatrices with near-zero determinant, we consider the random variable $\mathcal{W} = x_1y_2 - x_2y_1$, where x_1, x_2, y_1 and y_2 are the entries of a 2×2 submatrix.

By independence, $E(x_1y_2) = E(x_1)E(y_2)$ and $Var(x_1y_2) = Var(x_1)\,Var(y_2) + E(y_2)^2\,Var(x_1) + E(x_1)^2\,Var(y_2) = \sigma^4 + 2\mu^2\sigma^2$, and similarly for x_2y_1. Further, since x_2y_1 and x_2y_1 are independent,

$$E(x_1y_2 - x_2y_1) = E(x_1y_2) - E(x_2y_1) = 0 \text{ and}$$
$$Var(x_1y_2 - x_2y_1) = Var(x_1y_2) + Var(x_2y_1) = 2\sigma^4 + 4\mu^2\sigma^2.$$

Chebyshev's inequality [23] then implies that:

$$P(|\mathcal{W}| \geq \delta_{init}) \leq \frac{2\sigma^4 + 4\mu^2\sigma^2}{\delta_{init}^2},$$

giving a bound on the probability that a 2×2-submatrix deviates significantly from rank 1. The preliminary experiments presented in Appendix E of the extended version of the paper [5] additionally provide an empirical investigation of this probability. Assumptions on $\mathcal{Z}$ may lead to tighter bounds, a question that we leave open for future work.

6.3 Computational Complexity

Finally, we discuss the computational complexity of SampleAndExpand.

Consider a single iteration of the method. The runtime bottleneck is due to finding a maximum-edge biclique, which, in the worst case can take exponential time. However, the algorithm introduced by Lyu et al. [18] prunes large portions of the search space and can be very efficient in practice. As discussed in Section 5, to improve scalability, we can use a more scalable heuristic for finding an approximate maximum-edge biclique. The spectral biclustering algorithm, which is the heuristic we rely on by default, has computational complexity determined by the computation of the (truncated) SVD, which is $\mathcal{O}\left(\min(n^2m, m^2n)\right)$. If an even more scalable heuristic is leveraged, such as a basic linear-time algorithm removing rows and columns of $\mathbf{I}$ with less than a given proportion of ones, SampleAndExpand for the rank-1 case and for the general rank-k case incurs computational complexity $\mathcal{O}(n + m)$ and $\mathcal{O}(nkm)$, respectively.

As SampleAndExpand generally explores different initializations, if τ is the complexity of a single iteration, then $\mathcal{O}(N_{init}\tau)$ is the overall complexity.

7 Experiments

In this section, we evaluate the performance of SampleAndExpand against existing approaches. We consider both synthetic data and real-world data. More details on the experimental setup and additional experimental results are presented in the extended version of the paper [5].

Table 1: Summary characteristics for real-world datasets. We report the number of rows, columns, the low-rankness score, the entry-wise maximum squared deviation from the rank-1 SVD (Max rank-1 deviation) and a reference.

Dataset	# Rows	# Columns	Low-rankness score	Max rank-1 deviation	Reference
HYPERSPECTRAL	5 554	2 151	0.89	0.23	[17]
MOVIELENS	943	1682	0.30	1.00	[13]
CAMERAMAN	256	256	0.86	0.56	[24]

7.1 Experimental Setup

Datasets. We conduct experiments on both synthetic and real-world datasets.

The synthetic data are generated by planting near-rank-1 submatrices into larger matrices. To make the discovery task as challenging as possible, the entries of the planted submatrices and of the background are generated from the same distributions. We consider 6 different distributions. The details of the synthetic datasets are in Appendix D of the extended version of the paper [5].

Additionally, we consider 15 real-world matrices from different applications, including user ratings, images, and gene expression. We report summary characteristics for three datasets in Table 1, while the same information for the other datasets is provided in the extended version of the paper [5].

Baselines. We compare SAMPLEANDEXPAND against baselines discussed in Section 2. Specifically, we consider a method (CVX) based on convex optimization [8], PCA with sparsity constraints (SPARSEPCA) [20], SVP [22], and RPSP [6]. In the experiments with real data, we restrict the comparison to the most recently introduced methods, SVP and RPSP, which specifically aim at discovering (possibly multiple) near-low-rank submatrices.

Metrics. In experiments with synthetic data, all methods output matrices $\hat{\mathbf{D}}$ that contain low-rank approximations of the identified submatrices and zero entries for all indices that are not part of such submatrices. To measure the ability of a method in recovering the indices of the planted ground-truth submatrices, we report the F_1 score. Based on the same output, we also report the error (squared Frobenius norm averaged over the entries) incurred in approximating the ground-truth submatrices. SAMPLEANDEXPAND approximates submatrices through the interpretable approach discussed in Section 5.1 for $k = 1$ and via SVD for $k > 1$. All baselines approximate submatrices via SVD.

In real-world datasets, where no ground truth is available, we report the size and low-rankness score (introduced in Section 3) of the returned submatrices.

In all cases, we measure runtimes in seconds.

Parameters. The important parameter to set for our method is the tolerance δ controlling the trade-off between low-rankness and size. As explained in Section 6, one can set δ to match an input bound ϵ on the allowed low-rank-approximation error. In our experiments, however, we explore few fixed values of δ. Specifically,

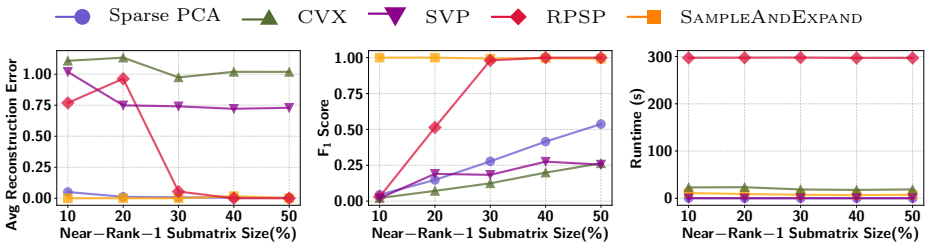

Fig. 5: Full-rank synthetic 250×250 matrices generated from a standard normal distribution with a planted near-rank-1 submatrix. Performance of different methods in the task of near-rank-1 submatrix discovery. We show the average (per-entry) reconstruction error (left), the F_1 score (center) and the runtime (right) of different methods as a function of planted submatrix size.

for experiments with synthetic data, we set δ to 0.05 and the number of initializations N_{init} to 25. For experiments with real-world data, we let δ vary in $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, and we consider $N_{init} = 25$ initializations for each value of δ. Finally, the initialization parameter δ_{init} is set to 10^{-11} and is increased by 10 every 10000 samples that do not result in a submatrix to expand.

Implementation. Our Python implementation and datasets are available online[4]. Experiments are performed on a computer with 2×10 core Xeon E5 processor and 256 GB memory. All reported results are averages over 10 runs.

7.2 Experiment Results

We first present results for synthetic datasets and then for real-world datasets.

Results on synthetic data. Figure 5 presents results for the task of near-rank-1-submatrix discovery in 250×250 matrices of entries generated from the standard normal distribution. Figure 5 in the extended version of the paper [5] provides analogous results for other 5 distributions. The results show that our method, unlike the baselines, consistently recovers the ground truth (as indicated by F_1 score close to 1 and reconstruction error close to 0). More specifically, RPSP tends to recover the ground-truth submatrix as its size increases, but it is also considerably slower than the other methods. SparsePCA and SVP are the fastest algorithms, but, like CVX, they often fail in detecting the ground truth.

Figure 6 shows, for data matrices with entries generated from a standard normal distribution, the same metrics as in Figure 5, but in the setting where multiple, possibly overlapping, near-rank-1 submatrices are planted and discovered. Figure 6 in the extended version of the paper [5] shows the same results for matrices generated from other 5 distributions. The results in this more challenging setting highlight that SampleAndExpand is the only method that consistently

[4] `https://github.com/maciap/SaE`

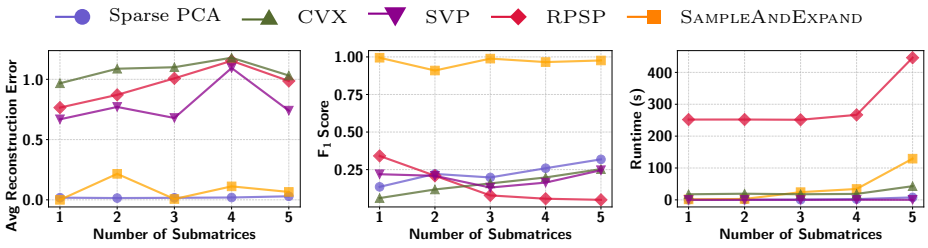

Fig. 6: Full-rank synthetic 250×250 matrices generated from a standard normal distribution with multiple (possibly overlapping) planted near-rank-1 submatrices. Performance of different methods in discovering planted submatrices. We show the average (per-entry) reconstruction error (left), the F_1 score (center) and the runtime (right) as a function of the number of planted submatrices.

retrieves the ground-truth submatrices. Among the baselines, SparsePCA stands out for its accurate reconstruction. However, the estimate $\hat{\mathbf{D}}$ of the input matrix it generates quickly becomes very dense as more submatrices are discovered, and hence this approach fails to identify the locations of the ground-truth submatrices.

Finally, Figure 10 in the extended version of the paper [5] (Appendix E) demonstrates the robustness of our method to the presence of noise.

Results on real-world data. Table 2 reports low-rankness score and size averaged over the top-5 submatrices retrieved by our method, SVP and RPSP for three datasets. Similar results for the other 12 datasets considered in our experiments are given in the extended version of the paper [5]. To determine the top-5 submatrices returned by each method, we select those that maximize the minimum between the low-rankness score and the size. Moreover, to offer a more complete picture, in the extended version of the paper [5] (Figure 11), we additionally display the low-rankness and size of the individual top-5 patterns.

Finding submatrices with high low-rankness is not an easy task. SVP returns large submatrices. However, those submatrices usually have smaller low-rankness compared to those discovered by our method or RPSP, and, in several cases, compared to the input matrix. SampleAndExpand and RPSP are more likely than SVP to return submatrices with large low-rankness. Further, SampleAndExpand tends to discover submatrices that strike a more desirable balance between low-rankness and size compared to RPSP. As concerns runtime, SampleAndExpand is drastically faster than RPSP in smaller datasets, but it can become slower in larger datasets. Nonetheless, the runtime of our method could be significantly reduced by leveraging a more efficient approach to maximum-edge-biclique extraction and by reducing the number of iterations, which, however, could deteriorate the quality of the results. As mentioned in Section 5.4, future work will consider efficient heuristic approaches to maximum-edge-biclique extraction.

Table 2: Performance in real-world data. For the top 5 local low-rank patterns identified by the methods, we show the average relative percentage increase (L-R) with respect to the low-rankness score of the input matrix, the size (in percentage of entries of the input matrix) and the runtime (in seconds) to obtain them.

Dataset	SVP			RPSP			SampleAndExpand		
	L-R	Size	Runtime	L-R	Size	Runtime	L-R	Size	Runtime
Hyperspectral	4.05	2.88	47.0	9.84	2.10	590	11.09	21.43	1 697
MovieLens	49.61	8.59	1.0	46.69	2.88	373	116.42	1.06	152
Cameraman	1.69	3.64	0.3	5.67	3.44	102	14.44	24.83	18

Finally, for our method, we also explore the trade-off between size and low-rankness by varying the value of δ; the results are presented in Appendix E of the extended version of the paper [5].

8 Conclusion

Low-rank approximation finds applications in many data-analysis tasks. Typically, methods assume that the entire matrix exhibits low-rank structure, while in real-world data this is often true only for certain submatrices. In this work, we study the problem of finding submatrices that are provably close to a rank-k approximation. We introduce a novel method that finds such submatrices, study the properties of the method, and, with a thorough experimental evaluation, we show that our method outperforms strong baselines.

There are several directions for future work. For instance, future work could study a more robust initialization strategy, develop more efficient and scalable alternative algorithms, and optimize the selection of the anchor rows and columns. It would also be valuable to investigate more the probabilistic aspects of our method. From a practical perspective, it would be interesting to explore further the benefits of our approach in different applications.

Acknowledgments. Martino Ciaperoni is supported by the European Union through the ERC-2018-ADG GA 834756 ("XAI: Science and Technology for the Explanation of AI Decision-Making") and the Partnership Extended PE00000013 ("FAIR: Future Artificial Intelligence Research"), Spoke 1: "Human-Centered AI". Aristides Gionis is supported by the ERC Advanced Grant REBOUND (834862), EC H2020 RIA project SoBigData++ (871042), and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Heikki Mannila is supported by the Technology Industries of Finland Centennial Foundation.

References

1. Agarwal, P.K., Procopiuc, C.M.: Approximation algorithms for projective clustering. Journal of Algorithms **46**(2), 115–139 (2003)

2. Boley, M., Lucchese, C., Paurat, D., Gärtner, T.: Direct local pattern sampling by efficient two-step random procedures. In: ACM SIGKDD. pp. 582–590 (2011)
3. Boutsidis, C., Mahoney, M.W., Drineas, P.: An improved approximation algorithm for the column subset selection problem. In: SODA. pp. 968–977 (2009)
4. Ciaperoni, M., Gionis, A., Mannila, H.: The Hadamard decomposition problem. Data Mining and Knowledge Discovery pp. 1–42 (2024)
5. Ciaperoni, M., Gionis, A., Mannila, H.: Sample and expand: Discovering low-rank submatrices with quality guarantees (2025), https://arxiv.org/abs/2506.06456
6. Dang, P., et al.: Generalized matrix local low rank representation by random projection and submatrix propagation. In: ACM SIGKDD. pp. 390–401 (2023)
7. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: ACM SIGKDD. pp. 269–274 (2001)
8. Doan, X.V., Vavasis, S.: Finding approximately rank-one submatrices with the nuclear norm and l_1-norm. SIAM Journal on Optimization **23**(4), 2502–2540 (2013)
9. Gillis, N., Glineur, F.: Using underapproximations for sparse nonnegative matrix factorization. Pattern recognition **43**(4), 1676–1687 (2010)
10. Gillis, N., Shitov, Y.: Low-rank matrix approximation in the infinity norm. Linear Algebra and its Applications **581**, 367–382 (2019)
11. Golub, G.H., Van Loan, C.F.: Matrix computations. JHU press (2013)
12. Guo, Q., Zhang, C., Zhang, Y., Liu, H.: An efficient SVD-based method for image denoising. IEEE transactions on Circuits and Systems for Video Technology **26**(5), 868–880 (2015)
13. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)
14. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022)
15. Kluger, Y., Basri, R., Chang, J.T., Gerstein, M.: Spectral biclustering of microarray data: coclustering genes and conditions. Genome research **13**(4), 703–716 (2003)
16. Lee, J., Kim, S., Lebanon, G., Singer, Y., Bengio, S.: Llorma: Local low-rank matrix approximation. Journal of Machine Learning Research **17**(15), 1–24 (2016)
17. Leone, G., et al.: Hyperspectral reflectance dataset of pristine, weathered and biofouled plastics. Earth System Science Data Discussions **2022**, 1–24 (2022)
18. Lyu, B., Qin, L., Lin, X., Zhang, Y., Qian, Z., Zhou, J.: Maximum biclique search at billion scale. Proceedings of the VLDB Endowment (2020)
19. Mahoney, M.W., Drineas, P.: Cur matrix decompositions for improved data analysis. Proceedings of the National Academy of Sciences **106**(3), 697–702 (2009)
20. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: ICML. pp. 689–696 (2009)
21. Miettinen, P., Neumann, S.: Recent developments in boolean matrix factorization. In: IJCAI (2021)
22. Ruchansky, N., Crovella, M., Terzi, E.: Targeted matrix completion. In: SIAM SDM. pp. 255–263 (2017)
23. Saw, J.G., Yang, M.C., Mo, T.C.: Chebyshev inequality with estimated mean and variance. The American Statistician **38**(2), 130–132 (1984)
24. University of Southern California, S., Institute, I.P.: Usc-sipi image database (2024), https://sipi.usc.edu/database/
25. Vidal, R.: Subspace clustering. IEEE Signal Processing **28**(2), 52–68 (2011)
26. Wang, Y., Zhu, L.: Research and implementation of SVD in machine learning. In: ICIS. pp. 471–475 (2017)