Parameterized Low-Rank Binary Matrix Approximation

Fedor V. Fomin
Department of Informatics, University of Bergen, Norway
Fedor.Fomin@uib.no
https://orcid.org/0000-0003-1955-4612

Petr A. Golovach
Department of Informatics, University of Bergen, Norway
Petr.Golovach@uib.no
https://orcid.org/0000-0002-2619-2990

Fahad Panolan
Department of Informatics, University of Bergen, Norway
Fahad.Panolan@uib.no
https://orcid.org/0000-0001-6213-8687

Abstract

We provide a number of algorithmic results for the following family of problems: For a given binary $m \times n$ matrix $\mathbf{A}$ and a nonnegative integer k, decide whether there is a "simple" binary matrix $\mathbf{B}$ which differs from $\mathbf{A}$ in at most k entries. For an integer r, the "simplicity" of $\mathbf{B}$ is characterized as follows.

- Binary r-Means: Matrix $\mathbf{B}$ has at most r different columns. This problem is known to be NP-complete already for $r = 2$. We show that the problem is solvable in time $2^{\mathcal{O}(k \log k)} \cdot (nm)^{\mathcal{O}(1)}$ and thus is fixed-parameter tractable parameterized by k. We also complement this result by showing that when being parameterized by r and k, the problem admits an algorithm of running time $2^{\mathcal{O}(r^{3/2} \cdot \sqrt{k \log k})}(nm)^{\mathcal{O}(1)}$, which is subexponential in k for $r \in o((k/\log k)^{1/3})$.
- Low GF(2)-Rank Approximation: Matrix $\mathbf{B}$ is of GF(2)-rank at most r. This problem is known to be NP-complete already for $r = 1$. It is also known to be W[1]-hard when parameterized by k. Interestingly, when parameterized by r and k, the problem is not only fixed-parameter tractable, but it is solvable in time $2^{\mathcal{O}(r^{3/2} \cdot \sqrt{k \log k})}(nm)^{\mathcal{O}(1)}$, which is subexponential in k for $r \in o((k/\log k)^{1/3})$.
- Low Boolean-Rank Approximation: Matrix $\mathbf{B}$ is of Boolean rank at most r. The problem is known to be NP-complete for $k = 0$ as well as for $r = 1$. We show that it is solvable in subexponential in k time $2^{\mathcal{O}(r2^r \cdot \sqrt{k \log k})}(nm)^{\mathcal{O}(1)}$.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms, Theory of computation → Fixed parameter tractability

Keywords and phrases Binary matrices, clustering, low-rank approximation, fixed-parameter tractability

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.53

Related Version A full version of the paper is available at [22], https://arxiv.org/abs/1803.06102.

Funding The research leading to these results have been supported by the Research Council of Norway via the projects "CLASSIS" and "MULTIVAL".

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 53; pp. 53:1–53:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements We thank Daniel Lokshtanov, Syed Mohammad Meesum and Saket Saurabh for helpful discussions on the topic of the paper.

1 Introduction

In this paper we consider the following generic problem. Given a binary $m \times n$ matrix, that is a matrix with entries from domain $\{0,1\}$, $\mathbf{A} = (a_{ij}) \in \{0,1\}^{m\times n}$, the task is to find a "simple" binary $m \times n$ matrix $\mathbf{B}$ which approximates $\mathbf{A}$ subject to some specified constrains. One of the most widely studied error measures is the *Frobenius norm*, which for a matrix $\mathbf{A}$ is defined as

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2}.$$

Here the sums are taken over $\mathbb{R}$. Then for a given nonnegative integer k, we want to decide whether there is a matrix $\mathbf{B}$ with certain properties such that $\|\mathbf{A}-\mathbf{B}\|_F^2 \le k$.

We consider the binary matrix approximation problems when for a given integer r, the approximation binary matrix $\mathbf{B}$

(A1) has at most r pairwise-distinct columns,
(A2) is of $\mathsf{GF}(2)$-rank at most r, and
(A3) is of Boolean rank at most r.

Each of these variants is very well-studied. Before defining each of the problems formally and providing an overview of the relevant results, the following observation is in order. Since we approximate a binary matrix by a binary matrix, in this case minimizing the Frobenius norm of $\mathbf{A}-\mathbf{B}$ is equivalent to minimizing the ℓ_0-norm of $\mathbf{A}-\mathbf{B}$, where the measure $\|\mathbf{A}\|_0$ is the number of non-zero entries of matrix $\mathbf{A}$. We also will be using another equivalent way of measuring the quality of approximation of a binary matrix $\mathbf{A}$ by a binary matrix $\mathbf{B}$ by taking the sum of the Hamming distances between their columns. Let us recall that the *Hamming distance* between two vectors $\mathbf{x},\mathbf{y} \in \{0,1\}^m$, where $\mathbf{x} = (x_1,\ldots,x_m)^\intercal$ and $\mathbf{y} = (y_1,\ldots,y_m)^\intercal$, is $d_H(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} |x_i - y_i|$ or, in words, the number of positions $i \in \{1,\ldots,m\}$ where x_i and y_i differ. Then for binary $m \times n$ matrix $\mathbf{A}$ with columns $\mathbf{a}^1,\ldots,\mathbf{a}^n$ and matrix $\mathbf{B}$ with columns $\mathbf{b}^1,\ldots,\mathbf{b}^n$, we define $d_H(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{n} d_H(\mathbf{a}^i,\mathbf{b}^i)$. In other words, $d_H(\mathbf{A},\mathbf{B})$ is the number of positions with different entries in matrices $\mathbf{A}$ and $\mathbf{B}$. Then we have the following.

$$\|\mathbf{A}-\mathbf{B}\|_F^2 = \|\mathbf{A}-\mathbf{B}\|_0 = d_H(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{n} d_H(\mathbf{a}^i,\mathbf{b}^i). \quad (1)$$

Problem (A1): Binary r-Means. By (1), the problem of approximating a binary $m \times n$ matrix $\mathbf{A}$ by a binary $m \times n$ matrix $\mathbf{B}$ with at most r different columns (problem (A1)) is equivalent to the following clustering problem. For given a set of n binary m-dimensional vectors $\mathbf{a}^1,\ldots,\mathbf{a}^n$ (which constitute the columns of matrix $\mathbf{A}$) and a positive integer r, Binary r-Means aims to partition the vectors in at most r clusters, as to minimize the sum of within-clusters sums of Hamming distances to their binary means. More formally,

Binary r-Means

Input: An $m \times n$ matrix $\mathbf{A}$ with columns $(\mathbf{a}^1,\ldots,\mathbf{a}^n)$, $r \in \mathbb{N}$ and a nonnegative integer k.
Question: Is there a positive integer $r' \le r$, a partition $\{I_1,\ldots,I_{r'}\}$ of $\{1,\ldots,n\}$ and vectors $\mathbf{c}^1,\ldots,\mathbf{c}^{r'} \in \{0,1\}^m$ such that $\sum_{i=1}^{r'}\sum_{j\in I_i} d_H(\mathbf{c}^i,\mathbf{a}^j) \le k$?

To see the equivalence of Binary r-Means and problem (A1), it is sufficient to observe that the pairwise different columns of an approximate matrix $\mathbf{B}$ such that $\|\mathbf{A}-\mathbf{B}\|_0 \le k$ can be used as vectors $\mathbf{c}^1, \dots, \mathbf{c}^{r'}$, $r' \le r$. As far as the mean vectors are selected, a partition of columns of $\mathbf{A}$ can be obtained by assigning each column-vector $\mathbf{a}^i$ to its closest mean vector $\mathbf{c}^j$ (ties breaking arbitrarily). Then for such clustering the total sum of distances from vectors within cluster to their centers does not exceed k. Similarly, a solution to Binary r-Means can be used as columns (with possible repetitions) of matrix $\mathbf{B}$ such that $\|\mathbf{A}-\mathbf{B}\|_0 \le k$. For that we put $\mathbf{b}^i = \mathbf{c}^j$, where $\mathbf{c}^j$ is the closest vector to $\mathbf{a}^i$.

This problem was introduced by Kleinberg, Papadimitriou, and Raghavan [37] as one of the examples of segmentation problems. Approximation algorithms for optimization versions of this problem were given by Alon and Sudakov [3] and Ostrovsky and Rabani [53], who referred to it as clustering in the Hamming cube. In bioinformatics, the case when $r = 2$ is known under the name Binary-Constructive-MEC (Minimum Error Correction) and was studied as a model for the Single Individual Haplotyping problem [13].

Binary r-Means can be seen as a discrete variant of the well-known k-Means Clustering. (Since in problems (A2) and (A3) we use r for the rank of the approximation matrix, we also use r in (A1) to denote the number of clusters which is commonly denoted by k in the literature on means clustering.) This problem has been studied thoroughly, particularly in the areas of computational geometry and machine learning. We refer to [1, 6, 39] for further references to the works on k-Means Clustering.

Problem (A2): Low GF(2)-Rank Approximation. Let $\mathbf{A}$ be a $m \times n$ binary matrix. In this case we view the elements of $\mathbf{A}$ as elements of GF(2), the Galois field of two elements. Then the GF(2)-rank of $\mathbf{A}$ is the minimum r such that $\mathbf{A} = \mathbf{U} \cdot \mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are $m \times r$ and $r \times n$ binary matrices respectively, and arithmetic operations are over GF(2). Equivalently, this is the minimum number of binary vectors, such that every column (row) of $\mathbf{A}$ is a linear combination (over GF(2)) of these vectors. Then (A2) is the following problem.

Low GF(2)-Rank Approximation

Input: An $m \times n$-matrix $\mathbf{A}$ over GF(2), $r \in \mathbb{N}$ and a nonnegative integer k.

Question: Is there a binary $m \times n$-matrix $\mathbf{B}$ with GF(2)-rank $\le r$ and $\|\mathbf{A}-\mathbf{B}\|_F^2 \le k$?

Low GF(2)-Rank Approximation arises naturally in applications involving binary data sets and serves as an important tool in dimension reduction for high-dimensional data sets with binary attributes, see [17, 35, 31, 38, 54, 57, 62] for further references and numerous applications of the problem.

Low GF(2)-Rank Approximation can be rephrased as a special variant (over GF(2)) of the problem finding the rigidity of a matrix. (For a target rank r, the *rigidity* of a matrix A over a field $\mathbb{F}$ is the minimum Hamming distance between A and a matrix of rank at most r.) Rigidity is a classical concept in Computational Complexity Theory studied due to its connections with lower bounds for arithmetic circuits [29, 30, 58, 55]. We refer to [41] for an extensive survey on this topic.

Low GF(2)-Rank Approximation is also a special case of a general class of problems approximating a matrix by a matrix with a small non-negative rank. Already Non-negative Matrix Factorization (NMF) is a nontrivial problem and it appears in many settings. In particular, in machine learning, approximation by a non-negative low rank matrix has gained extreme popularity after the influential article in Nature by Lee and Seung [40]. NMF is an ubiquitous problem and besides machine learning, it has been independently introduced and studied in combinatorial optimization [21, 61], and communication complexity [2, 42]. An extended overview of applications of NMF in statistics, quantum mechanics, biology, economics, and chemometrics, can be found in the work of Cohen and Rothblum [15] and recent books [12, 51, 25].

Problem (A3): Low Boolean-Rank Approximation. Let $\mathbf{A}$ be a binary $m \times n$ matrix. This time we view the elements of $\mathbf{A}$ as *Boolean* variables. The *Boolean rank* of $\mathbf{A}$ is the minimum r such that $\mathbf{A} = \mathbf{U} \wedge \mathbf{V}$ for a Boolean $m \times r$ matrix $\mathbf{U}$ and a Boolean $r \times n$ matrix $\mathbf{V}$, where the product is Boolean, that is, the logical $\wedge$ plays the role of multiplication and $\vee$ the role of sum. Here $0 \wedge 0 = 0$, $0 \wedge 1 = 0$, $1 \wedge 1 = 1$, $0 \vee 0 = 0$, $0 \vee 1 = 1$, and $1 \vee 1 = 1$. Thus the matrix product is over the Boolean semi-ring $(0, 1, \wedge, \vee)$. This can be equivalently expressed as the normal matrix product with addition defined as $1+1 = 1$. Binary matrices equipped with such algebra are called *Boolean matrices*. Equivalently, $\mathbf{A} = (a_{ij}) \in \{0,1\}^{m \times n}$ has the Boolean rank 1 if $\mathbf{A} = \mathbf{x}^\intercal \wedge \mathbf{y}$, where $\mathbf{x} = (x_1, x_2, \ldots, x_m) \in \{0,1\}^m$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \{0,1\}^n$ are nonzero vectors and the product is Boolean, that is, $a_{ij} = x_i \wedge y_j$. Then the Boolean rank of $\mathbf{A}$ is the minimum integer r such that $\mathbf{A} = \mathbf{a^{(1)}} \vee \cdots \vee \mathbf{a^{(r)}}$, where $\mathbf{a^{(1)}}, \ldots, \mathbf{a^{(r)}}$ are matrices of Boolean rank 1; zero matrix is the unique matrix with the Boolean rank 0. Then Low Boolean-Rank Approximation is defined as follows.

Low Boolean-Rank Approximation

Input: A Boolean $m \times n$ matrix $\mathbf{A}$, $r \in \mathbb{N}$ and a nonnegative integer k.
Question: Is there a Boolean $m \times n$ matrix $\mathbf{B}$ of Boolean rank $\leq r$ and $d_H(\mathbf{A}, \mathbf{B}) \leq k$?

For $r = 1$ Low Boolean-Rank Approximation coincides with Low GF(2)-Rank Approximation but for $r > 1$ these are different problems.

Boolean low-rank approximation has attracted much attention, especially in the data mining and knowledge discovery communities. In data mining, matrix decompositions are often used to produce concise representations of data. Since much of the real data is binary or even Boolean in nature, Boolean low-rank approximation could provide a deeper insight into the semantics associated with the original matrix. There is a big body of work done on Low Boolean-Rank Approximation, see e.g. [7, 9, 17, 43, 48, 49].

P-Matrix Approximation. While at first glance Low GF(2)-Rank Approximation and Low Boolean-Rank Approximation look very similar, algorithmically the latter problem is more challenging. The fact that GF(2) is a field allows to play with different equivalent definitions of rank like row rank and column ranks. We exploit this strongly in our algorithm for Low GF(2)-Rank Approximation. For Low Boolean-Rank Approximation the matrix product is over the Boolean semi-ring and nice properties of the GF(2)-rank cannot be used here (see, e.g. [32]). Our algorithm for Low Boolean-Rank Approximation is based on solving an auxiliary $\mathbf{P}$-Matrix Approximation problem, where the task is to approximate a matrix $\mathbf{A}$ by a matrix $\mathbf{B}$ whose block structure is defined by a given pattern matrix $\mathbf{P}$. It appears, that $\mathbf{P}$-Matrix Approximation is also an interesting problem on its own.

More formally, let $\mathbf{P} = (p_{ij}) \in \{0,1\}^{p \times q}$ be a binary $p \times q$ matrix. We say that a binary $m \times n$ matrix $\mathbf{B} = (b_{ij}) \in \{0,1\}^{m \times n}$ is a $\mathbf{P}$-*matrix* if there is a partition $\{I_1, \ldots, I_p\}$ of $\{1, \ldots, m\}$ and a partition $\{J_1, \ldots, J_q\}$ of $\{1, \ldots, n\}$ such that for every $i \in \{1, \ldots, p\}$, $j \in \{1, \ldots, q\}$, $s \in I_i$ and $t \in J_j$, $b_{st} = p_{ij}$. In words, the columns and rows of $\mathbf{B}$ can be permuted such that the block structure of the resulting matrix is defined by $\mathbf{P}$.

P-Matrix Approximation

Input: An $m \times n$ binary matrix $\mathbf{A}$, a pattern binary matrix $\mathbf{P}$ and a nonnegative integer k.
Question: Is there an $m \times n$ $\mathbf{P}$-matrix $\mathbf{B}$ such that $\|\mathbf{A}-\mathbf{B}\|_F^2 \le k$?

The notion of **P**-matrix was implicitly defined by Wulff et al. [60] as an auxiliary tool for their approximation algorithm for the related monochromatic biclustering problem. Since Low GF(2)-Rank Approximation remains NP-complete for $r = 1$ [26], we have that **P**-Matrix Approximation is NP-complete already for the very simple pattern matrix $P = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$.

1.1 Related work

In this subsection we give an overview of previous related algorithmic and complexity results for problems (A1)–(A3), as well as related problems. Since each of the problems has many practical applications, there is a tremendous amount of literature on heuristics and implementations. In this overview we concentrate on known results about algorithms with proven guarantee, with emphasis on parameterized complexity.

Problem (A1): Binary r-Means. Binary r-Means is trivially solvable in polynomial time for $r = 1$, and as was shown by Feige in [20], is NP-complete for every $r \ge 2$.

PTAS (polynomial time approximation scheme) for optimization variants of Binary r-Means were developed in [3, 53]. Approximation algorithms for more general k-Means Clustering is a thoroughly studied topic [1, 6, 39]. Inaba et al. [33] have shown that the general k-Means Clustering is solvable in time n^{mr+1} (here n is the number of vectors, m is the dimension and r the number of required clusters). We are not aware of any, except the trivial brute-force, exact algorithm for Binary r-Means prior to our work.

Problem (A2): Low GF(2)-Rank Approximation. When the low-rank approximation matrix **B** is not required to be binary, then the optimal Frobenius norm rank-r approximation of (not necessarily binary) matrix **A** can be efficiently found via the singular value decomposition (SVD). This is an extremely well-studied problem and we refer to surveys for an overview of algorithms for low rank approximation [36, 44, 59]. However, SVD does not guarantee to find an optimal solution in the case when additional structural constrains on the low-rank approximation matrix **B** (like being non-negative or binary) are imposed.

In fact, most of these constrained variants of low-rank approximation are NP-hard. In particular, Gillis and Vavasis [26] and Dan et al. [17] have shown that Low GF(2)-Rank Approximation is NP-complete for every $r \ge 1$. Approximation algorithms for the optimization version of Low Boolean-Rank Approximation were considered in [34, 35, 17, 38, 57, 10] among others.

Most of the known results about the parameterized complexity of the problem follows from the results for Matrix Rigidity. Fomin et al. have proved in [24] that for every finite field, and in particular GF(2), Matrix Rigidity over a finite field is W[1]-hard being parameterized by k. This implies that Low GF(2)-Rank Approximation is W[1]-hard when parameterized by k. However, when parameterized by k and r, the problem becomes fixed-parameter tractable. For Low GF(2)-Rank Approximation, the algorithm from [24] runs in time $2^{\mathcal{O}(f(r)\sqrt{k}\log k)}(nm)^{\mathcal{O}(1)}$, where f is some function of r. While the function $f(r)$ is not specified in [24], the algorithm in [24] invokes enumeration of all $2^r \times 2^r$ binary matrices of rank r, and thus the running time is at least double-exponential in r.

Meesum, Misra, and Saurabh [46], and Meesum and Saurabh [47] considered parameterized algorithms for related problems about editing of the adjacencies of a graph (or directed graph) targeting a graph with adjacency matrix of small rank.

Problem (A3): Low Boolean-Rank Approximation. It follows from the rank definitions that a matrix is of Boolean rank $r = 1$ if and only if its $\mathsf{GF}(2)$-rank is 1. Thus by the results of Gillis and Vavasis [26] and Dan et al. [17] Low Boolean-Rank Approximation is NP-complete already for $r = 1$.

While computing $\mathsf{GF}(2)$-rank (or rank over any other field) of a matrix can be performed in polynomial time, deciding whether the Boolean rank of a given matrix is at most r is already an NP-complete problem. Thus Low Boolean-Rank Approximation is NP-complete already for $k = 0$. This follows from the well-known relation between the Boolean rank and covering edges of a bipartite graph by bicliques [28]. Let us briefly describe this equivalence. For Boolean matrix $\mathbf{A}$, let G_A be the corresponding bipartite graph, i.e. the bipartite graph whose biadjacency matrix is $\mathbf{A}$. By the equivalent definition of the Boolean rank, $\mathbf{A}$ has Boolean rank r if and only if it is the logical disjunction of r Boolean matrices of rank 1. But for every bipartite graph whose biadjacency matrix is a Boolean matrix of rank at most 1, its edges can be covered by at most one biclique (complete bipartite graph). Thus deciding whether a matrix is of Boolean rank r is exactly the same as deciding whether edges of a bipartite graph can be covered by at most r bicliques. The latter Biclique Cover problem is known to be NP-complete [52]. Biclique Cover is solvable in time $2^{2^{\mathcal{O}(r)}}(nm)^{\mathcal{O}(1)}$ [27] and unless Exponential Time Hypothesis (ETH) fails, it cannot be solved in time $2^{2^{o(r)}}(nm)^{\mathcal{O}(1)}$ [11].

For the special case $r = 1$ Low Boolean-Rank Approximation and $k \leq \|A\|_0/240$, Bringmann, Kolev and Woodruff gave an exact algorithm of running time $2^{k/\sqrt{\|A\|_0}}(nm)^{\mathcal{O}(1)}$ [10]. (Let us remind that the ℓ_0-norm of a matrix is the number of its non-zero entries.) More generally, exact algorithms for NMF were studied by Cohen and Rothblum in [15]. Arora et al. [5] and Moitra [50], who showed that for a fixed value of r, NMF is solvable in polynomial time. Related are also the works of Razenshteyn et al. [56] on weighted low-rank approximation, Clarkson and Woodruff [14] on robust subspace approximation, and Basu et al. [8] on PSD factorization.

Observe that all these problems could be seen as matrix editing problems. For Binary r-Means, we can assume that $r \leq n$ as otherwise we have a trivial YES-instance. Then the problem asks whether it is possible to edit at most k entries of the input matrix, that is, replace some 0-s by 1-s and some 1-s by 0-s, in such a way that the obtained matrix has at most r pairwise-distinct columns. Respectively, Low GF(2)-Rank Approximation asks whether it is possible to edit at most k entries of the input matrix to obtain a matrix of rank at most r. In $\mathbf{P}$-Matrix Approximation, we ask whether we can edit at most k elements to obtain a $\mathbf{P}$-matrix. A lot of work in graph algorithms has been done on graph editing problems, in particular parameterized subexponential time algorithms were developed for a number of problems, including various cluster editing problems [19, 23].

1.2 Our results and methods

We study the parameterized complexity of Binary r-Means, Low GF(2)-Rank Approximation and Low Boolean-Rank Approximation. We refer to the recent books of Cygan et al. [16] and Downey and Fellows [18] for the introduction to Parameterized Algorithms and Complexity. Our results are summarized in Table 1.

Table 1 Parameterized complexity of low-rank approximation. GF(2) Approx stands for Low GF(2)-Rank Approximation and Bool Approx for Low Boolean-Rank Approximation. We omit the polynomial factor $(nm)^{\mathcal{O}(1)}$ in running times.

	k	r	$k+r$
Binary r-Means	$2^{\mathcal{O}(k\log k)}$, Thm 1	NP-c for $r \geq 2$ [20]	$2^{\mathcal{O}(r^{3/2}\cdot\sqrt{k\log k})}$, Thm 3
GF(2) Approx	W[1]-hard [24]	NP-c for $r \geq 1$ [26, 17]	$2^{\mathcal{O}(r^{3/2}\cdot\sqrt{k\log k})}$, Thm 4
Boolean Approx	NP-c for $k = 0$ [52]	NP-c for $r \geq 1$ [26, 17]	$2^{\mathcal{O}(r2^{r}\cdot\sqrt{k\log k})}$, Thm 2

Our first main result concerns Binary r-Means. We show (Theorem 1) that the problem is solvable in time $2^{\mathcal{O}(k\log k)}\cdot(nm)^{\mathcal{O}(1)}$. Therefore, Binary r-Means is FPT parameterized by k. Since Low GF(2)-Rank Approximation parameterized by k is W[1]-hard and Low Boolean-Rank Approximation is NP-complete for any fixed $k \geq 0$, we find Theorem 1 quite surprising. The proof of Theorem 1 is based on a fundamental result of Marx [45] about the complexity of a problem on strings, namely Consensus Patterns. We solve Binary r-Means by constructing a two-stage FPT Turing reduction to Consensus Patterns. First, we use the color coding technique of Alon, Yuster, and Zwick from [4] to reduce Binary r-Means to some special auxiliary problem and then show that this problem can be reduced to Consensus Patterns, and this allows us to apply the algorithm of Marx [45].

Our second main result concerns Low Boolean-Rank Approximation. As we mentioned above, the problem is NP-complete for $k = 0$, as well as for $r = 1$, and hence is intractable being parameterized by k or by r only. On the other hand, a simpler Low GF(2)-Rank Approximation is not only FPT parameterized by $k+r$, by [24] it is solvable in time $2^{\mathcal{O}(f(r)\sqrt{k}\log k)}(nm)^{\mathcal{O}(1)}$, where f is some function of r, and thus is *subexponential* in k. It is natural to ask whether a similar complexity behavior could be expected for Low Boolean-Rank Approximation. Our second main result, Theorem 2, shows that this is indeed the case: Low Boolean-Rank Approximation is solvable in time $2^{\mathcal{O}(r2^{r}\cdot\sqrt{k\log k})}(nm)^{\mathcal{O}(1)}$. The proof of this theorem is technical and consists of several steps. We first develop a subexponential algorithm for solving auxiliary **P**-Matrix Approximation, and then construct an FPT Turing reduction from Low Boolean-Rank Approximation to **P**-Matrix Approximation.

Let us note that due to the relation of Boolean rank computation to Biclique Cover, the result of [11] implies that unless Exponential Time Hypothesis (ETH) fails, Low Boolean-Rank Approximation cannot be solved in time $2^{2^{o(r)}}f(k)(nm)^{\mathcal{O}(1)}$ for any function f. Thus the dependence in r in our algorithm cannot be improved significantly unless ETH fails.

Interestingly, the technique developed for solving **P**-Matrix Approximation can be used to obtain algorithms of running times $2^{\mathcal{O}(r^{3/2}\cdot\sqrt{k\log k})}(nm)^{\mathcal{O}(1)}$ for Binary r-Means and Low GF(2)-Rank Approximation (Theorems 3 and 4 respectively). For Binary r-Means, Theorems 3 provides much better running time than Theorem 1 for values of $r \in o((k\log k)^{1/3})$.

For Low GF(2)-Rank Approximation, comparing Theorem 4 and the running time $2^{\mathcal{O}(f(r)\sqrt{k}\log k)}(nm)^{\mathcal{O}(1)}$ from [24], let us note that Theorem 4 not only slightly improves the exponential dependence in k by $\sqrt{\log k}$; it also drastically improves the exponential dependence in r, from $2^{2^{r}}$ to $2^{r^{3/2}}$.

Due to space restrictions, we only give high level descriptions of our algorithms. In Section 2 we sketch the algorithm for Binary r-Means parameterized by k, and in Section 3 we explain how we construct FPT algorithms for Binary r-Means and Low GF(2)-Rank Approximation parameterized by k and r that are subexponential in k. The full proofs and further results can be found in [22].

2 Binary r-Means parameterized by k

In this section we give a description of our FPT algorithm for Binary r-Means that runs in time $2^{\mathcal{O}(k \log k)} \cdot (nm)^{\mathcal{O}(1)}$ (Theorem 1).

Let $(\mathbf{A}, r, k)$ be an instance of Binary r-Means where $\mathbf{A}$ is a matrix with columns $(\mathbf{a}^1, \dots, \mathbf{a}^n)$. We say that a partition $\{I_1, \dots, I_{r'}\}$ of $\{1, \dots, n\}$ for $r' \leq r$ is a *solution* for this instance if there are vectors $\mathbf{c}^1, \dots, \mathbf{c}^{r'} \in \{0,1\}^m$ such that $\sum_{i=1}^{r'} \sum_{j \in I_i} d_H(\mathbf{c}^i, \mathbf{a}^j) \leq k$. We say that each I_i or, equivalently, the multiset of columns $\{\mathbf{a}^j \mid j \in I_i\}$ (some columns could be the same) is a *cluster* and $\mathbf{c}^i$ the *mean* of the cluster. Observe that given a cluster $I \subseteq \{1, \dots, n\}$, one can easily compute an optimal mean $\mathbf{c} = (c_1, \dots, c_m)^\intercal$ as follows. Let $\mathbf{a}^j = (a_{1j}, \dots, a_{mj})^\intercal$ for $j \in \{1, \dots, n\}$. For each $i \in \{1, \dots, m\}$, consider the multiset $S_i = \{a_{ij} \mid j \in I\}$ and put $c_i = 0$ or $c_i = 1$ according to the *majority* of elements in S_i, that is, $c_i = 0$ if at least half of the elements in S_i are 0-s and $c_i = 1$ otherwise. We refer to this construction of $\mathbf{c}$ as the *majority rule*.

An *initial cluster* is an inclusion maximal set $I \subseteq \{1, \dots, n\}$ such that all the columns in the initial cluster are equal. The property of initial clusters we build upon is that there is always an optimal solution that does not split any of the initial clusters. More formally, we say that a partition $\{I_1, \dots, I_{r'}\}$ of the columns of matrix $\mathbf{A}$ is *regular* if for every initial cluster I, there is $i \in \{1, \dots, r'\}$ such that $I \subseteq I_i$. Respectively, if $(\mathbf{A}, r, k)$ is a yes-instance of Binary r-Means, then there is a solution $\{I_1, \dots, I_{r'}\}$ forming a regular partition and we call such a solution *regular*. In words, in a regular solution every two equal columns of $\mathbf{A}$ are placed in the same cluster. Thus in a regular solution $\{I_1, \dots, I_{r'}\}$, each cluster I_i is either *simple*, that is, contains exactly one initial cluster and all its columns are equal, or I_i is *composite*, that is composed of several initial clusters. One can show that there is always an optimal solution to Binary r-Means that is regular.

Moreover, we can assume that a solution we seek for is not only regular but has stronger property. Indeed, let $\{I_1, \dots, I_{r'}\}$ be a regular solution for instance $(\mathbf{A}, r, k)$. Denote by $\mathbf{c}^1, \dots, \mathbf{c}^{r'}$ the corresponding means of the clusters. Let I_i be a composite cluster of $\{I_1, \dots, I_{r'}\}$ that contains $h \geq 2$ initial clusters. Then $\sum_{j \in I_i} (\mathbf{c}^i, \mathbf{a}^j) \geq h - 1$. Therefore, for every yes-instance, a regular solution $\{I_1, \dots, I_{r'}\}$ contains at most k composite clusters; all the remaining clusters are simple. Moreover, the total number of initial clusters used to form the composite clusters is at most $2k$. Note also that if I_i is a simple cluster then for $\mathbf{c}^i = \mathbf{a}^h$ for an arbitrary $h \in I_i$, we have that $\sum_{j \in I_i} (\mathbf{c}^i, \mathbf{a}^j) = 0$, that is, simple clusters do not contribute to the total cost of the solution.

Thus the essence of the problem is to find the way of composing initial clusters into composite ones. More precisely, for the instance $(\mathbf{A}, r, k)$, let $\mathcal{I}$ be the family of the initial clusters. Let $s = |\mathcal{I}|$. Then finding a solution for Binary r-Means is equivalent to finding a set $\mathcal{I}' \subseteq \mathcal{I}$ of size at most $2k$ such that $\mathcal{I}'$ can be used to form at most $r - s + |\mathcal{I}'|$ composite clusters. In other words, we are looking for $\mathcal{I}' \subseteq \mathcal{I}$ of size at most $2k$ such that there is a partition $\{P_1, \dots, P_t\}$ of $\mathcal{I}'$ with $t \leq r - s + |\mathcal{I}'|$ (each set P_i says which initial clusters of $\mathcal{I}'$ are used to form a composite cluster) and vectors $\mathbf{s}^1, \dots, \mathbf{s}^t \in \{0,1\}^m$ with the property that

$$\sum_{i=1}^{t} \sum_{I \in P_i} \sum_{j \in I} d_H(\mathbf{s}^i, \mathbf{a}^j) \leq k.$$

If $s \leq r$, then $(\mathbf{A}, r, k)$ is a trivial yes-instance of the problem with $\mathcal{I}$ being a solution. If $r + k < s$, then $(\mathbf{A}, r, k)$ is a trivial no-instance. From now on we assume that $r < s \leq r + k$.

If we color uniformly at random initial clusters with $2k$ colors, then with a “reasonable” probability each composite cluster is composed from initial clusters of different colors. This will provide us with an additional structural information, which will bring us much closer to a solution. We use the classic *color coding* technique of Alon, Yuster, and Zwick from [4] to distinguish initial clusters of $\mathcal{I}'$ from each other. At the end we obtain a deterministic algorithm but, for simplicity, we describe a randomized Monte-Carlo algorithm here. We color the elements of $\mathcal{I}$ independently and uniformly at random by $2k$ colors $1, \ldots, 2k$. Observe that if $(\mathbf{A}, r, k)$ is a yes-instance, then at most $2k$ initial clusters in a solution that are included in composite clusters are colored by distinct colors with the probability at least $\frac{(2k)!}{(2k)^{2k}} \geq e^{-2k}$. We say that a solution $\{I_1, \ldots, I_{r'}\}$ for $(\mathbf{A}, r, k)$ is a *colorful* solution if all initial clusters that are included in composite clusters of $\{I_1, \ldots, I_{r'}\}$ are colored by distinct colors. We construct an algorithm for finding a colorful solution (if it exists).

Let us fix some coloring of initial clusters in $2k$ colors. For $i \in \{1, \ldots, 2k\}$, let $\mathcal{I}_i$ be the set of initial clusters colored by color i. Note that some sets $\mathcal{I}_i$ could be empty. We consider all possible partitions $\mathcal{P} = \{P_1, \ldots, P_t\}$ of nonempty subsets of $\{1, \ldots, 2k\}$ such that each set of $\mathcal{P}$ contains at least two elements. Notice that if $(\mathbf{A}, r, k)$ has a colorful solution $\{I_1, \ldots, I_{r'}\}$, then there is $\mathcal{P} = \{P_1, \ldots, P_t\}$ such that a cluster I_i of the solution is formed from initial clusters colored with colors P_j for some $j \in \{1, \ldots, t\}$. Moreover, two different composite clusters are colored with colors from different sets of $\mathcal{P}$. We go through all possible partitions $\mathcal{P}$, and if $(\mathbf{A}, r, k)$ has a colorful solution, we will find the corresponding partition $\mathcal{P}$. Let us fix a partition $\mathcal{P} = \{P_1, \ldots, P_t\}$. If $s - |P_1| - \cdots - |P_t| + t > r$, we discard the current choice of $\mathcal{P}$. Assume from now that this is not the case. For each $i \in \{1, \ldots, t\}$, we do the following. Let $P_i = \{i_1, \ldots, i_p\} \subseteq \{1, \ldots, 2k\}$. Then with this notation $\mathcal{I}_{i_j}$ is the set of initial clusters colored by color i_j. For $j \in \{1, \ldots, p\}$, we use $J^i_j = \bigcup_{I \in \mathcal{I}_{i_j}} I$ to denote the set of indices contained in clusters colored by i_j. We also define $J^i = J^i_1 \cup \cdots \cup J^i_p$, which is the set of indices contained in clusters colored by colors from $\{i_1, \ldots, i_p\}$. Denote by $\mathbf{A}_i$ the submatrix of $\mathbf{A}$ containing the columns $\mathbf{a}^h$ with $h \in J^i$. We want to solve an auxiliary problem of finding the minimum integer $d_i \leq k$ such that there is a set of initial clusters $L^i_1, \ldots, L^i_p$ and a vector $\mathbf{s}^i \in \{0,1\}^m$ such that $L^i_j \subseteq J^i_j$ for $i \in \{1, \ldots, p\}$ and $\sum_{i=1}^{p} \sum_{j \in L^i_j} d_H(\mathbf{s}^i, \mathbf{a}^j) \leq d_i$. In words, for a set of colors $P_i = \{i_1, \ldots, i_p\}$, we want to find the best selection of initial clusters $L^i_1, \ldots, L^i_p$ such that each of the clusters L^i_j is colored by i_j; the best is in the sense that the total Hamming distance d_i from the columns corresponding to the selected set of clusters to their means is the minimum over all such selections.

Assume that we have an algorithm for this auxiliary problem. If such a value of d_i does not exist for some $i \in \{1, \ldots, t\}$, we discard the current choice of $\mathcal{P}$. Otherwise, we find the set of clusters $L^i_1, \ldots, L^i_p$ and $\mathbf{s}^i$. Let $L^i = L^i_1 \cup \cdots \cup L^i_p$. We check whether $d_1 + \cdots + d_t \leq k$. If it holds, we return the colorful solution with the composite clusters $L^1, \ldots, L^t$ whose means are $\mathbf{s}^1, \ldots, \mathbf{s}^t$ respectively and the remaining clusters are simple. Otherwise, we discard the choice of $\mathcal{P}$. If for one of the choices of $\mathcal{P}$ we find a colorful solution, we return it and stop. If we fail to find a solution for all possible $\mathcal{P}$, we return the answer NO and stop. If the described algorithm produces a solution, then because simple clusters do not contribute to the total cost of the solution it is possible to verify that the produced solution is a colorful solution to $(\mathbf{A}, r, k)$.

So everything boils down to finding the optimal value d_i together with the corresponding initial clusters and their means. Let us remind that a regular partition $\{I_1, \ldots, I_p\}$ of $\{1, \ldots, n\}$ is a partition where any two indices corresponding to equal columns of $\mathbf{A}$ are assigned to the same cluster I_i. We call this the auxiliary problem Cluster Selection.

Cluster Selection

Input: An $m \times n$-matrix $\mathbf{A}$ with columns $\mathbf{a}^1, \ldots, \mathbf{a}^n$, a regular partition $\{I_1, \ldots, I_p\}$ of $\{1, \ldots, n\}$, and a nonnegative integer d.

Question: Is there a set of initial clusters $J_1, \ldots, J_p$ and a vector $\mathbf{c} \in \{0,1\}^m$ such that $J_i \subseteq I_i$ for $i \in \{1, \ldots, p\}$ and $\sum_{i=1}^{p} \sum_{j \in J_i} d_H(\mathbf{c}, \mathbf{a}^j) \leq d$?

Thus in Cluster Selection, for each cluster I_i, we have to select exactly one initial cluster J_i contained in I_i such that the total Hamming distance of the columns of the selected clusters to their means does not exceed d.

We prove that Cluster Selection is FPT when parameterized by d. The proof of this result is based on a reduction to the problem about strings. More precisely, we apply the result of Marx [45] about the Consensus Patterns problem. Recall that for two strings a and b of the same length, the *Hamming distance* $d_H(a, b)$ between strings is defined as the number of position where the strings differ.

Consensus Patterns

Input: A set of p strings $\{s_1, \ldots, s_p\}$ over an alphabet Σ, a positive integer t and a nonnegative integer d.

Question: Is there a string s of length t over Σ, and a length t substring s'_i of s_i for every $i \in \{1, \ldots, p\}$ such that $\sum_{i=1}^{p} d_H(\mathbf{s}, s'_i) \leq d$?

Marx proved in [45] that Consensus Patterns can be solved in time $\delta^{\mathcal{O}(\delta)} \cdot |\Sigma|^{\delta} \cdot L^9$ where $\delta = d/p$ and L the input size, i.e., the total length of all the strings in the input. This implies that Consensus Patterns can be solved in time $2^{\mathcal{O}(d \log d)} \cdot L^9$ if $|\Sigma|$ is fixed. We construct an FPT Turing reduction from Cluster Selection to Consensus Patterns. The reduction is technical but the rough idea is the following. We guess the number of elements in every cluster of the solution $J_1, \ldots, J_p$. For each guess, $(\ell_1 = |J_1|, \ldots, \ell_p = |J_p|)$, we delete from I_i all initial clusters which size is not equal to ℓ_i. Then for each I_i, we make ℓ_i equal strings. Each of theses strings consists of substrings corresponding to distinct columns of $\mathbf{A}$ with indices from I_i which are separated by special splitting substrings constructed by making use of two additional symbols. Thus in total we use alphabet Σ with 4 letters. This way, the choice of substrings corresponds to the choice of initial clusters.

Summarizing, our algorithm for Binary r-Means consists of two FPT Turing reductions. First, we design a reduction to Cluster Selection that, in its turn, is reduced to Consensus Patterns. This gives our first main result.

▶ **Theorem 1.** Binary r-Means *is solvable in time* $2^{\mathcal{O}(k \log k)} \cdot (nm)^{\mathcal{O}(1)}$.

3 Subexponential algorithms

The main result of this section is an FPT algorithm for Low Boolean-Rank Approximation parameterized by k and r that is subexponential in k. Note that Low Boolean-Rank Approximation is more complicated than Low GF(2)-Rank Approximation. The main reason to that is that the elements of the matrices do not form a field and thus many nice properties of matrix-rank cannot be used here. The way we handle this issue is to solve the **P**-Matrix Approximation problem.

Let $\mathbf{A}$ be a Boolean $m \times n$-matrix with the Boolean rank $r \geq 1$. Then $\mathbf{A} = \mathbf{A}^{(1)} \vee \ldots \vee \mathbf{A}^{(r)}$ where $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(r)}$ are matrices of Boolean rank 1. It implies that $\mathbf{A}$ has at most 2^r pairwise-distinct rows and at most 2^r pairwise-distinct columns. Hence, the Boolean rank of $\mathbf{A}$ is at most r if and only if there is a $2^r \times 2^r$-matrix $\mathbf{P}$ of Boolean rank at most r such that

$\mathbf{A}$ is a $\mathbf{P}$-matrix. Respectively, the Low Boolean-Rank Approximation problem can be reformulated as follows: Decide whether there is a $2^r \times 2^r$-pattern matrix $\mathbf{P}$ with the Boolean rank at most r and an $m \times n$ $\mathbf{P}$-matrix $\mathbf{B}$ such that $\|\mathbf{A}-\mathbf{B}\|_F^2 \le k$. We generate all $2^r \times 2^r$-matrices $\mathbf{P}$ of Boolean rank at most r, and then for each matrix $\mathbf{P}$, we solve $\mathbf{P}$-Matrix Approximation for the instance $(\mathbf{A}, \mathbf{P}, k)$. We return YES if we obtain at least one yes-instance of $\mathbf{P}$-Matrix Approximation, and we return NO otherwise. Thus, using the algorithm for $\mathbf{P}$-Matrix Approximation from Theorem 5 (See later in this section), we obtain the following theorem.

▶ **Theorem 2.** Low Boolean-Rank Approximation *is solvable in* $2^{\mathcal{O}(r2^r\sqrt{k\log k})}\cdot(nm)^{\mathcal{O}(1)}$ *time.*

Now we sketch the main ideas behind our algorithm for $\mathbf{P}$-Matrix Approximation. In fact, we exploit the same ideas to construct subexponential in k time algorithms for Binary r-Means and Low GF(2)-Rank Approximation, and these algorithms are great deal less technical. Hence, we concentrate here on these problems.

Let $(\mathbf{A}, r, k)$ be an instance of Binary r-Means where $\mathbf{A}$ is a matrix with columns $(\mathbf{a}^1, \ldots, \mathbf{a}^n)$. Recall that given a solution $\{I_1, \ldots, I_{r'}\}$, one can compute the corresponding means $\mathbf{c}^1, \ldots, \mathbf{c}^{r'} \in \{0,1\}^m$ using the majority rule. Note that in the opposite direction, given a set of means $\mathbf{c}^1, \ldots, \mathbf{c}^{r'}$, we can construct clusters $\{I_1, \ldots, I_{r'}\}$ as follows: for each column $\mathbf{a}^j$, find the closest $\mathbf{c}^i$, $i \in \{1, \ldots, r'\}$, that is such that $d_H(\mathbf{c}^i, \mathbf{a}^j)$ is minimum and assign j to I_i. Note that this procedure does not guarantee that all clusters are nonempty but we can simply delete empty clusters. Hence, we can define a solution as a set of means $C = \{\mathbf{c}^1, \ldots, \mathbf{c}^{r'}\}$.

It can be observed that we can restrict ourself by considering only solutions of special type. Let $\mathbf{a}_1, \ldots, \mathbf{a}_m$ be the rows of $\mathbf{A}$. We say that a vector $\mathbf{c} = (c_1, \ldots, c_m)^\intercal \in \{0,1\}^m$ *agrees* with $\mathbf{A}$ if $c_i = c_j$ whenever $\mathbf{a}_i = \mathbf{a}_j$ for $i, j \in \{1, \ldots, m\}$. If $(\mathbf{A}, r, k)$ is a yes-instance of Binary r-Means, then it can be shown that $(\mathbf{A}, r, k)$ has a solution such that for each cluster of the solution its mean agrees with $\mathbf{A}$. Also it could be seen that if $(\mathbf{A}, r, k)$ is a yes-instance of Binary r-Means, then $\mathbf{A}$ has at most $r + k$ pairwise-distinct columns and at most $2^r + k$ pairwise-distinct rows. These observations allow us to construct a recursive branching algorithm for Binary r-Means.

First, we preprocess the instance $(\mathbf{A}, r, k)$: if $\mathbf{A}$ has at least $r + k + 1$ pairwise-distinct columns or at least $2^r + k + 1$ distinct rows, we return the answer NO and stop. Now on we assume that this is not the case.

Assume that we are given a partial clustering of some columns of $\mathbf{A}$ represented by a family of means $\{\mathbf{c}_1, \ldots, \mathbf{c}_s\}$, a budget d and the set I of remaining columns of $\mathbf{A}$ which have to be clustered. The algorithm tries to extend the partial solution by not exceeding the budget $d \le k$. Some of the columns from I can go to the existing cluster and some can form new clusters. Suppose that we know the minimum Hamming distance $h \le d$ from vectors in new clusters to their means (in the algorithm we consider all possible values of h). Then all vectors which are within distance less than h to the already existing means, can be assigned to the existing clusters. Then we will be basically left with two options. Either the number of columns to be assigned to new clusters does not exceed $\sqrt{d\log(2^r+d)} \le \sqrt{k\log(2^r+k)}$; in this case we brute-force in all possible partitions of I. Or we can upper bound $h \le \sqrt{d/\log(2^r+d)} \le \sqrt{k/\log(2^r+k)}$. In the latter case we branch on all possible vectors that agree with $\mathbf{A}$ and are at distance at most h from one of the at most $r + k$ columns of I. Due to the fact that the number of distinct rows of $\mathbf{A}$ is at most $2^r + k$, the number of branches for each column of I is at most $(2^r + k)^h$. In each branch of our algorithm, we add a new mean to the partial solution and call our algorithm recursively. Note that the depth of the recursion is upper bounded by r.

This way we obtain the following theorem.

▶ **Theorem 3.** Binary r-Means *is solvable in time* $2^{\mathcal{O}(r\sqrt{k\log(2^r+k)})}\cdot(nm)^{\mathcal{O}(1)}$.

The general idea of the subexponential time algorithm for Low GF(2)-Rank Approximation is similar but the algorithm is more complicated. Let $(\mathbf{A},r,k)$ be an instance of Low GF(2)-Rank Approximation where $\mathbf{a}^1,\ldots,\mathbf{a}^n$ are the columns of $\mathbf{A}$. We observe that $(\mathbf{A},r,k)$ is a yes-instance if and only if there is a positive integer $r'\leq r$ and linearly independent vectors $\mathbf{c}^1,\ldots,\mathbf{c}^{r'}\in\{0,1\}^m$ over $\mathsf{GF}(2)$ such that

$$\sum_{i=1}^{n}\min\{d_H(\mathbf{s},\mathbf{a}^i)\mid \mathbf{s}=\bigoplus_{j\in I}\mathbf{c}^j,\ I\subseteq\{1,\ldots,r'\}\}\leq k;$$

we use "$\bigoplus$" to denote the summation over $\mathsf{GF}(2)$. Respectively, we construct the recursive branching algorithm for Low GF(2)-Rank Approximation that tries to extend a partial solution represented by a family of linearly independent vectors. We use the properties that if $(\mathbf{A},r,k)$ is a yes-instance, then the number of pairwise distinct columns and rows is at most 2^r+k and we can select new vectors that agree with $\mathbf{A}$. We obtain the following theorem.

▶ **Theorem 4.** Low GF(2)-Rank Approximation *is solvable in time* $2^{\mathcal{O}(r\sqrt{k\log(2^r+k)})}\cdot(nm)^{\mathcal{O}(1)}$.

For $\mathbf{P}$-Matrix Approximation, we use the same approach based on the combination of branching and local search but because we have to follow the structure of the pattern matrix P, the algorithm becomes technical. We get the following running time for the problem.

▶ **Theorem 5.** $\mathbf{P}$-Matrix Approximation *is solvable in time* $2^{\mathcal{O}((p+q)\sqrt{k\log(p+k)}+p\log p+q\log q+q\log(q+k))}\cdot(nm)^{\mathcal{O}(1)}$.

Note that the running time in Theorem 5 is asymmetric in p and q due to the fact that we treat rows and columns in different way but, trivially, the instances $(\mathbf{A},\mathbf{P},k)$ and $(\mathbf{A}^\intercal,\mathbf{P}^\intercal,k)$ of $\mathbf{P}$-Matrix Approximation are equivalent.

4 Conclusion and open problems

In this paper we provide a number of parameterized algorithms for a number of binary matrix-approximation problems. Our results uncover some parts of the complexity landscape of these fascinating problems. We hope that our work will facilitate further investigation of this important and exciting area. We conclude with the following concrete open problems about bivariate complexity of Binary r-Means, Low GF(2)-Rank Approximation, and Low Boolean-Rank Approximation.

For Binary r-Means we have shown that the problem is solvable in time $2^{\mathcal{O}(k\log k)}\cdot(nm)^{\mathcal{O}(1)}$. A natural question is whether this running time is optimal. While the lower bound of the kind $2^{o(k)}\cdot(nm)^{\mathcal{O}(1)}$ or $2^{o(k\log k)}\cdot(nm)^{\mathcal{O}(1)}$ seems to be most plausible here, we do not know any strong argument against, say a $2^{o(k)}\cdot(nm)^{\mathcal{O}(1)}$-time algorithm. At least for the number of distinct columns $r\in o((k/\log k)^{1/3})$ we have a subexponential in k algorithm, so maybe we can solve the problem in time subexponential in k for any value of r?

For Low GF(2)-Rank Approximation we have an algorithm solving the problem in time $2^{\mathcal{O}(r^{3/2}\cdot\sqrt{k\log k})}(nm)^{\mathcal{O}(1)}$. Here, shaving off the $\sqrt{\log k}$ factor in the exponent seems to be a reasonable thing. However, we do not know how to do it even at the cost of a worse dependence in r. In other words, could the problem be solvable in time $2^{\mathcal{O}(f(r)\cdot\sqrt{k})}(nm)^{\mathcal{O}(1)}$ for some function f? On the other hand, we also do not know how to rule out algorithms running in time $2^{o(r)\cdot o(k)}(nm)^{\mathcal{O}(1)}$.

For Low Boolean-Rank Approximation, how far is our upper bound $2^{\mathcal{O}(r2^r\cdot\sqrt{k\log k})}(nm)^{\mathcal{O}(1)}$ from the optimal? For example, we know that for any function f, the solvability of the problem in time $2^{2^{o(r)}}f(k)(nm)^{\mathcal{O}(1)}$ implies the failure of ETH. Could we rule out any $2^{o(\sqrt{k})}f(r)(nm)^{\mathcal{O}(1)}$ algorithm?

References

1 Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004. doi:10.1145/1008731.1008736.

2 Alfred V. Aho, Jeffrey D. Ullman, and Mihalis Yannakakis. On notions of information transfer in VLSI circuits. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 133–139. ACM, 1983. doi:10.1145/800061.808742.

3 Noga Alon and Benny Sudakov. On two segmentation problems. *J. Algorithms*, 33(1):173–184, 1999. doi:10.1006/jagm.1999.1024.

4 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

5 Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization - provably. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pages 145–162. ACM, 2012.

6 Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 250–257. ACM, 2002. doi:10.1145/509907.509947.

7 Eduard Bartl, Radim Belohlávek, and Jan Konecny. Optimal decompositions of matrices with grades into binary and graded matrices. *Annals of Mathematics and Artificial Intelligence*, 59(2):151–167, Jun 2010. doi:10.1007/s10472-010-9185-y.

8 Amitabh Basu, Michael Dinitz, and Xin Li. Computing approximate PSD factorizations. *CoRR*, abs/1602.07351, 2016. URL: http://arxiv.org/abs/1602.07351, arXiv:1602.07351.

9 Radim Belohlávek and Vilém Vychodil. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *J. Computer and System Sciences*, 76(1):3–20, 2010. doi:10.1016/j.jcss.2009.05.002.

10 Karl Bringmann, Pavel Kolev, and David P. Woodruff. Approximation algorithms for ℓ_0-low rank approximation. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 6651–6662, 2017. URL: http://papers.nips.cc/paper/7242-approximation-algorithms-for-ell_0-low-rank-approximation.

11 L. Sunil Chandran, Davis Issac, and Andreas Karrenbauer. On the parameterized complexity of biclique cover and partition. In *Proceedings of the 11th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 63 of *LIPIcs*, pages 11:1–11:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.11.

12 Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

13 Rudi Cilibrasi, Leo van Iersel, Steven Kelk, and John Tromp. The complexity of the single individual SNP haplotyping problem. *Algorithmica*, 49(1):13–36, 2007. doi:10.1007/s00453-007-0029-z.

14 Kenneth L. Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 310–329. IEEE Computer Society, 2015.

15 Joel E Cohen and Uriel G Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra and its Applications*, 190:149–168, 1993.

16 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

17 Chen Dan, Kristoffer Arnsfelt Hansen, He Jiang, Liwei Wang, and Yuchen Zhou. On low rank approximation of binary matrices. *CoRR*, abs/1511.01699, 2015. arXiv:1511.01699.

18 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

19 Pål Grønås Drange, Felix Reidl, Fernando Sanchez Villaamil, and Somnath Sikdar. Fast biclustering by dual parameterization. *CoRR*, abs/1507.08158, 2015.

20 Uriel Feige. NP-hardness of hypercube 2-segmentation. *CoRR*, abs/1411.0821, 2014. arXiv:1411.0821.

21 Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *J. ACM*, 62(2):17, 2015. doi:10.1145/2716307.

22 Fedor V. Fomin, Petr A. Golovach, and Fahad Panolan. Parameterized low-rank binary matrix approximation. *CoRR*, abs/1803.06102, 2018.

23 Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, and Yngve Villanger. Tight bounds for parameterized complexity of cluster editing with a small number of clusters. *J. Computer and System Sciences*, 80(7):1430–1447, 2014.

24 Fedor V. Fomin, Daniel Lokshtanov, Syed Mohammad Meesum, Saket Saurabh, and Meirav Zehavi. Matrix rigidity from the viewpoint of parameterized complexity. In *Proceedings of the 34th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 66 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:14, 2017. doi:10.4230/LIPIcs.STACS.2017.32.

25 Yun Fu. *Low-Rank and Sparse Modeling for Visual Analysis*. Springer International Publishing, 1 edition, 2014.

26 Nicolas Gillis and Stephen A. Vavasis. On the complexity of robust PCA and ℓ_1-norm low-rank matrix approximation. *CoRR*, abs/1509.09236, 2015. URL: http://arxiv.org/abs/1509.09236, arXiv:1509.09236.

27 Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Data reduction and exact algorithms for clique cover. *ACM Journal of Experimental Algorithmics*, 13, 2008. doi:10.1145/1412228.1412236.

28 David A. Gregory, Norman J. Pullman, Kathryn F. Jones, and J. Richard Lundgren. Biclique coverings of regular bigraphs and minimum semiring ranks of regular matrices. *J. Combinatorial Theory Ser. B*, 51(1):73–89, 1991. doi:10.1016/0095-8956(91)90006-6.

29 Dmitry Grigoriev. Using the notions of separability and independence for proving the lower bounds on the circuit complexity (in russian). *Notes of the Leningrad branch of the Steklov Mathematical Institute, Nauka*, 1976.

30 Dmitry Grigoriev. Using the notions of separability and independence for proving the lower bounds on the circuit complexity. *Journal of Soviet Math.*, 14(5):1450–1456, 1980.

31 Harold W. Gutch, Peter Gruber, Arie Yeredor, and Fabian J. Theis. ICA over finite fields - separability and algorithms. *Signal Processing*, 92(8):1796–1808, 2012. doi:10.1016/j.sigpro.2011.10.003.

32 Alexander E. Guterman. Rank and determinant functions for matrices over semirings. In *Surveys in contemporary mathematics*, volume 347 of *London Math. Soc. Lecture Note Ser.*, pages 1–33. Cambridge Univ. Press, Cambridge, 2008.

33 Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the 10th annual symposium on Computational Geometry*, pages 332–339. ACM, 1994.

34 Peng Jiang and Michael T. Heath. Mining discrete patterns via binary matrix factorization. In *ICDM Workshops*, pages 1129–1136. IEEE Computer Society, 2013.

35 Peng Jiang, Jiming Peng, Michael Heath, and Rui Yang. *A Clustering Approach to Constrained Binary Matrix Factorization*, pages 281–303. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

36 Ravindran Kannan and Santosh Vempala. Spectral algorithms. *Foundations and Trends in Theoretical Computer Science*, 4(3-4):157–288, 2009. doi:10.1561/0400000025.

37 Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Segmentation problems. *J. ACM*, 51(2):263–280, 2004. doi:10.1145/972639.972644.

38 Mehmet Koyutürk and Ananth Grama. Proximus: A framework for analyzing very high dimensional discrete-attributed datasets. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 147–156, New York, NY, USA, 2003. ACM. doi:10.1145/956750.956770.

39 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2):5:1–5:32, 2010. doi:10.1145/1667053.1667054.

40 Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

41 Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Found. Trends Theor. Comput. Sci.*, 4:1–155, 2009.

42 László Lovász and Michael E. Saks. Lattices, Möbius functions and communication complexity. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 81–90. IEEE, 1988.

43 Haibing Lu, Jaideep Vaidya, Vijayalakshmi Atluri, and Yuan Hong. Constraint-aware role mining via extended boolean matrix decomposition. *IEEE Trans. Dependable Sec. Comput.*, 9(5):655–669, 2012. doi:10.1109/TDSC.2012.21.

44 Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011. doi:10.1561/2200000035.

45 Dániel Marx. Closest substring problems with small distances. *SIAM J. Comput.*, 38(4):1382–1410, 2008. doi:10.1137/060673898.

46 S. M. Meesum, Pranabendu Misra, and Saket Saurabh. Reducing rank of the adjacency matrix by graph modification. *Theoret. Comput. Sci.*, 654:70–79, 2016. doi:10.1016/j.tcs.2016.02.020.

47 Syed Mohammad Meesum and Saket Saurabh. Rank reduction of directed graphs by vertex and edge deletions. In *Proceedings of the 12th Latin American Symposium on (LATIN)*, volume 9644 of *Lecture Notes in Comput. Sci.*, pages 619–633. Springer, 2016. doi:10.1007/978-3-662-49529-2_46.

48 Pauli Miettinen, Taneli Mielikäinen, Aristides Gionis, Gautam Das, and Heikki Mannila. The discrete basis problem. *IEEE Trans. Knowl. Data Eng.*, 20(10):1348–1362, 2008. doi:10.1109/TKDE.2008.53.

49 Pauli Miettinen and Jilles Vreeken. Model order selection for boolean matrix factorization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 51–59. ACM, 2011. doi:10.1145/2020408.2020424.

50 Ankur Moitra. An almost optimal algorithm for computing nonnegative rank. *SIAM J. Comput.*, 45(1):156–173, 2016. doi:10.1137/140990139.

51 Ganesh R Naik. *Non-negative Matrix Factorization Techniques.* Springer, 2016.

52 James Orlin. Contentment in graph theory: covering graphs with cliques. *Nederl. Akad. Wetensch. Proc. Ser. A* **80**=*Indag. Math.*, 39(5):406–424, 1977.

53 Rafail Ostrovsky and Yuval Rabani. Polynomial-time approximation schemes for geometric min-sum median clustering. *J. ACM*, 49(2):139–156, 2002. doi:10.1145/506147.506149.

54 Amichai Painsky, Saharon Rosset, and Meir Feder. Generalized independent component analysis over finite alphabets. *IEEE Trans. Information Theory*, 62(2):1038–1053, 2016. doi:10.1109/TIT.2015.2510657.

55 A A Razborov. On rigid matrices. *Manuscript in russian*, 1989.

56 Ilya P. Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 250–263. ACM, 2016. doi:10.1145/2897518.2897639.

57 Bao-Hong Shen, Shuiwang Ji, and Jieping Ye. Mining discrete patterns via binary matrix factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 757–766, New York, NY, USA, 2009. ACM. doi:10.1145/1557019.1557103.

58 Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Mathematical Foundations of Computer Science (MFCS)*, volume 53 of *Lecture Notes in Comput. Sci.*, pages 162–176. Springer, 1977.

59 David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014. doi:10.1561/0400000060.

60 Sharon Wulff, Ruth Urner, and Shai Ben-David. Monochromatic bi-clustering. In *Proceedings of the 30th International Conference on Machine Learning, (ICML)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 145–153. JMLR.org, 2013. URL: http://jmlr.org/proceedings/papers/v28/.

61 Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *J. Comput. Syst. Sci.*, 43(3):441–466, 1991. doi:10.1016/0022-0000(91)90024-Y.

62 Arie Yeredor. Independent component analysis over Galois fields of prime order. *IEEE Trans. Information Theory*, 57(8):5342–5359, 2011. doi:10.1109/TIT.2011.2145090.