Parameterized Complexity of Feature Selection for Categorical Data Clustering

Sayan Bandyapadhyay
Department of Informatics, University of Bergen, Norway

Fedor V. Fomin
Department of Informatics, University of Bergen, Norway

Petr A. Golovach
Department of Informatics, University of Bergen, Norway

Kirill Simonov
Algorithms and Complexity Group, TU Wien, Austria

Abstract

We develop new algorithmic methods with provable guarantees for feature selection in regard to categorical data clustering. While feature selection is one of the most common approaches to reduce dimensionality in practice, most of the known feature selection methods are heuristics. We study the following mathematical model. We assume that there are some inadvertent (or undesirable) features of the input data that unnecessarily increase the cost of clustering. Consequently, we want to select a subset of the original features from the data such that there is a small-cost clustering on the selected features. More precisely, for given integers ℓ (the number of irrelevant features) and k (the number of clusters), budget B, and a set of n categorical data points (represented by m-dimensional vectors whose elements belong to a finite set of values Σ), we want to select $m - \ell$ relevant features such that the cost of any optimal k-clustering on these features does not exceed B. Here the cost of a cluster is the sum of Hamming distances (ℓ_0-distances) between the selected features of the elements of the cluster and its center. The clustering cost is the total sum of the costs of the clusters.

We use the framework of parameterized complexity to identify how the complexity of the problem depends on parameters k, B, and $|\Sigma|$. Our main result is an algorithm that solves the Feature Selection problem in time $f(k, B, |\Sigma|) \cdot m^{g(k,|\Sigma|)} \cdot n^2$ for some functions f and g. In other words, the problem is fixed-parameter tractable parameterized by B when $|\Sigma|$ and k are constants. Our algorithm for Feature Selection is based on a solution to a more general problem, Constrained Clustering with Outliers. In this problem, we want to delete a certain number of outliers such that the remaining points could be clustered around centers satisfying specific constraints. One interesting fact about Constrained Clustering with Outliers is that besides Feature Selection, it encompasses many other fundamental problems regarding categorical data such as Robust Clustering, Binary and Boolean Low-rank Matrix Approximation with Outliers, and Binary Robust Projective Clustering. Thus as a byproduct of our theorem, we obtain algorithms for all these problems. We also complement our algorithmic findings with complexity lower bounds.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Mathematics of computing → Combinatorial algorithms

Keywords and phrases Robust clustering, PCA, Low rank approximation, Hypergraph enumeration

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.14

Related Version *Full Version*: `https://arxiv.org/abs/2105.03753`

Funding This work is supported by the Research Council of Norway via the project "MULTIVAL".

Acknowledgements The authors are thankful to the anonymous reviewers for their helpful comments.

1 Introduction

Clustering is one of the most fundamental concepts in data mining and machine learning. A considerable challenge to the clustering approaches is the high dimensionality of modern datasets. When the data contains many irrelevant features (or attributes), an application of cluster analysis with a complete set of features could significantly decrease the solution's quality. The typical approach to overcome this challenge in practice is *feature selection*. The method is based on selecting a small subset of relevant features from the data and applying the clustering algorithm only on the selected features. The survey of [1] provides a comprehensive overview on methods for feature selection in clustering. Due to the significance of feature selection, there is a multitude of heuristic methods addressing the problem. However, very few provably correct methods are known [6, 7, 11].

Kim et al. [20] introduced a model of feature selection in the context of k-means clustering. We use their motivating example here. Decision-making based on market surveys is a pragmatic marketing strategy used by manufacturers to increase customer satisfaction. The respondents of a survey are segmented into similar-interest groups so that each group of customers can be treated in a similar way. Consider such a market survey data that typically contains responses of customers to a set of questions regarding their demographic and psychographic information, shopping experience, attitude towards new products and expectations from the business. The standard practice used by market managers to segment customers is to apply clustering techniques w.r.t. the whole set of features. However, depending on the application, responses corresponding to some of the features might not be relevant to find the target set of market segments. Also, some of the responses might contain incomplete or spurious information. To address this issue, Kim et al. [20] considered several quality criteria to return *Pareto* optimal (or non-dominated) solutions that optimize one or more criteria. One such solution removes a suitable subset of features and cluster the data w.r.t. the remaining features.

The main objective of this work is to study clustering problems on *categorical* data. In statistics, a categorical variable is a variable that can admit a fixed number of possible values. For example, it could be a gender, blood type, political orientation, etc. A prominent example of categorical data is binary data where the points are vectors each of whose coordinates can take value either 0 or 1. Binary data arise in several important applications. In electronic commerce, each transaction can be modeled as a binary vector (known as market basket data) each of whose coordinates denotes whether a particular item is purchased or not [32, 22]. The most common similarity (or dissimilarity) measure for categorical data objects is the Hamming distance, which is basically the number of mismatched attributes of the objects.

2 Our results

In this paper, we introduce a new model of feature selection w.r.t. categorical data clustering, which is motivated by the work of Kim et al. [20]. We assume that there are some inadvertent features of the input data that unnecessarily increase the cost of clustering. Consequently, in our model, we define the best subset of features (of a given size) as the subset that minimizes the corresponding cost of clustering. The goal is to compute such a subset and the respective clusters. We provide the first parameterized algorithmic and complexity results for feature selection in regard to categorical data clustering.

Let Σ be a finite set of non-negative integers. We refer to Σ as the alphabet and we denote the m-dimensional space over Σ by Σ^m. Given two m-dimensional vectors $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^m$, the *Hamming distance* (or ℓ_0-*distance*) $d_H(\boldsymbol{x}, \boldsymbol{y})$ is the number of different coordinates in $\boldsymbol{x}$ and

$\boldsymbol{y}$, that is $d_H(\boldsymbol{x},\boldsymbol{y}) = |\{i \in \{1,\dots,m\} : \boldsymbol{x}[i] \neq \boldsymbol{y}[i]\}|$. For a set of indices $S \subset \{1,2,\dots,m\}$ and an $m \times n$ matrix $\boldsymbol{A}$, let $\boldsymbol{A}^{-S}$ be the matrix obtained from $\boldsymbol{A}$ by removing the rows with indices in S. We denote the columns of $\boldsymbol{A}^{-S}$ by $\boldsymbol{a}^j_{-S}$ for $1 \leq j \leq n$. We consider the following mathematical model of feature selection.

Feature Selection

Input: An alphabet Σ, an $m \times n$ matrix $\boldsymbol{A}$ with columns $\boldsymbol{a}^1, \boldsymbol{a}^2, \dots, \boldsymbol{a}^n$ such that $\boldsymbol{a}^j \in \Sigma^m$ for all $1 \leq j \leq n$, a positive integer k, non-negative integers B and ℓ.

Task: Decide whether there is a subset $O \subset \{1,2,\dots,m\}$ of size at most ℓ, a partition $\{I_1, I_2, \dots, I_k\}$ of $\{1,2,\dots,n\}$, and vectors $\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_k \in \Sigma^{m-|O|}$ such that

$$\sum_{i=1}^{k} \sum_{j \in I_i} d_H(\boldsymbol{a}^j_{-O}, \boldsymbol{c}_i) \leq B.$$

In the above definition and all subsequent problem definitions, without loss of generality, we assume that each cluster is non-empty, i.e., $I_i \neq \emptyset$ for each $1 \leq i \leq k$. Note that, otherwise, one could probe different values $k' < k$ for the actual number of non-empty clusters. The problem is defined as a decision problem, however if the instance is a yes-instance we would also like to find such a clustering. For $\ell = 0$, Feature Selection is the popular Binary k-Clustering problem, which is known to be NP-hard for every $k \geq 2$ [13]. This makes it natural to investigate the parameterized complexity of Feature Selection.

Our main result is the following theorem.

▶ **Theorem 1.** *Feature Selection is solvable in time $f(k, B, |\Sigma|) \cdot m^{g(k,|\Sigma|)} \cdot n^2$, where f and g are computable functions.*

In particular, this implies that for fixed k and $|\Sigma|$, the problem is fixed-parameter tractable (FPT[1]) parameterized by B. Note that in any study concerning the parameterized complexity of a problem, the value of the parameter is implicitly assumed to be sufficiently small compared to the input size. Although the parameter B seems to be a natural choice from the problem definition, in general B can be fairly large. Hence, Theorem 1 is mostly applicable in the scenario when for the selected features the cost of clustering B is small and thus the points are well-clustered on the selected features. Even in this case the problem is far from being trivial. One can think of our problem as a problem from the broader class of editing problems, where the goal is to check whether a given instance is close to a "structured" one. In particular, our problem can be seen as the problem of editing the input matrix after removing at most ℓ rows such that the resulting matrix contains at most k distinct columns and the number of edits does not exceed the budget B. In this sense, our work is in line with the work of [18] on matrix completion and [15] on clustering. Note that in many applications it is reasonable to assume that k and $|\Sigma|$ are bounded, as the alphabet size and the number of clusters are small in practice. Indeed, for binary data, $|\Sigma| = 2$.

Another interesting property of our algorithm is that the running time does not depend on the number of irrelevant features ℓ. In particular, for fixed k, B, and $|\Sigma|$, it runs in polynomial time even when $\ell = \Omega(m)$. Also, the theorem could be used to identify the minimum number of irrelevant features ℓ such that the cost of k-clustering on the remaining features does not exceed B. Note that our time complexity also exponentially depends on the

[1] A problem is FPT or fixed-parameter tractable parameterized by a set of parameters if it can be solved by algorithms that are exponential only in the values of the parameters while polynomial in the size of the input.

number of clusters k. In this regard, one can compare our result with the result in [15] that shows that the binary version of the problem with $\ell = 0$ (BINARY k-CLUSTERING) is FPT parameterized by B only. However, in the presence of irrelevant features, the dependence on k is unavoidable as we state in our next theorem.

▶ **Theorem 2.** *FEATURE SELECTION is* W[1]*-hard parameterized by*

- *either* $k + (m - \ell)$
- *or* ℓ

even when $B = 0$ *and* $\Sigma = \{0, 1\}$*. Moreover, assuming the Exponential Time Hypothesis (ETH), the problem cannot be solved in time* $f(k) \cdot m^{o(k)} \cdot n^{\mathcal{O}(1)}$ *for any function* f*, even when* $B = 0$ *and the alphabet* Σ *is binary.*

Note that when $B = 0$ and $\Sigma = \{0, 1\}$, from Theorem 1 it follows that FEATURE SELECTION can be solved in time $f(k) \cdot m^{g(k)} \cdot n^2$. Theorem 2 shows that the dependence of such a function g on k is inevitable, unless W[1] = FPT, and $g(k)$ is unlikely to be sublinear up to ETH.

In order to prove Theorem 1, we prove a more general theorem about CONSTRAINED CLUSTERING WITH OUTLIERS. In this problem, one seeks a clustering with centers of clusters satisfying the property imposed by a set of relations. Constrained clustering [14] was introduced as the tool in the design of approximation algorithms for binary low-rank approximation problems. The CONSTRAINED CLUSTERING WITH OUTLIERS problem is basically the robust variant of this problem. As we will see, by the reduction given in Lemma 5, Theorem 4 proves Theorem 1. Besides FEATURE SELECTION, CONSTRAINED CLUSTERING WITH OUTLIERS encompasses a number of well-studied problems around robust clustering, low-rank matrix approximation, and dimensionality reduction. Our algorithm for constrained clustering implies fixed-parameter tractability for all these problems.

To define constrained clustering, we need a few definitions. A *p-ary relation* on Σ is a collection of p-tuples whose elements are in Σ.

▶ **Definition 3** (Vectors satisfying $\mathcal{R}$). *An ordered set* $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_p\}$ *of* m*-dimensional vectors in* Σ^m *is said to satisfy a set* $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ *of* p*-ary relations on* Σ *if for all* $1 \le i \le m$*, the* p*-tuple formed by the* i*-th coordinates of vectors from* C*, that is* $(\boldsymbol{c}_1[i], \boldsymbol{c}_2[i], \ldots, \boldsymbol{c}_p[i])$*, belongs to* R_i*.*

We define the following constrained variant of robust categorical clustering.

CONSTRAINED CLUSTERING WITH OUTLIERS

Input: An alphabet Σ, an $m \times n$ matrix $\boldsymbol{A}$ with columns $\boldsymbol{a}^1, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^n$ such that $\boldsymbol{a}^j \in \Sigma^m$ for all $1 \le j \le n$, a positive integer k, non-negative integers B and ℓ, a set $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ of k-ary relations on Σ.

Task: Decide whether there is a subset $O \subset \{1, 2, \ldots, n\}$ of size at most ℓ, a partition $\mathcal{I} = \{I_1, I_2, \ldots, I_k\}$ of $\{1, 2, \ldots, n\} \setminus O$, and a set $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ of m-dimensional vectors in Σ^m such that C satisfies $\mathcal{R}$ and

$$\sum_{i=1}^{k} \sum_{j \in I_i} d_H(\boldsymbol{a}^j, \boldsymbol{c}_i) \le B.$$

Thus in CONSTRAINED CLUSTERING WITH OUTLIERS we want to identify a set of outliers $\boldsymbol{a}^i, i \in O$, such that the remaining $n - \ell$ vectors could be partitioned into k *clusters* $\{I_1, I_2, \ldots, I_k\}$. Each cluster I_j could be identified by its *center* $\boldsymbol{c}_j \in \Sigma^m$ as the set of vectors that are closer to $\boldsymbol{c}_j \in \Sigma^m$ than to any other center (ties are broken arbitrarily). Then the cost of each cluster I_j is the sum of the Hamming distances between its vectors and the corresponding center $\boldsymbol{c}_j \in \Sigma^m$. However, there is an additional condition that the set of cluster centers $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ must satisfy the set of k-ary relations $\mathcal{R}$. And, the total sum of costs of all clusters must not exceed B. We prove the following theorem.

▶ **Theorem 4.** *Constrained Clustering with Outliers is solvable in time*

$$(kB)^{O(kB)}|\Sigma|^{kB} \cdot n^{O(k)} \cdot m^2.$$

The connection between Constrained Clustering with Outliers and Feature Selection is established in the following lemma.

▶ **Lemma 5.** *For any instance $(\boldsymbol{D}, k, B, \ell)$ of Feature Selection, one can construct in time $\mathcal{O}(mn + k \cdot |\Sigma|^k)$ an equivalent instance $(\boldsymbol{A}, k', B', \ell', \mathcal{R})$ of Constrained Clustering with Outliers such that $\boldsymbol{A}$ is the transpose of $\boldsymbol{D}$, $k' = |\Sigma|^k$, $B' = B$ and $\ell' = \ell$.*

Theorem 1 follows from Theorem 4 and Lemma 5. Connections of constrained clustering with several other clustering and low-rank matrix approximation problems have been established in the literature [14]. Similarly, Theorem 4 allows to design parameterized algorithms for robust variants of these problems.

Robust low-rank matrix approximation. Here we discuss two problems where for a given matrix of categorical data, we seek to remove ℓ columns such that the remaining columns are well approximated by a matrix of small rank. The vanilla case of the ℓ_0-Low Rank Approximation problem is the following. Given an $m \times n$ matrix $\boldsymbol{A}$ over GF(p) (a finite field of order p), the task is to find an $m \times n$ matrix $\boldsymbol{B}$ over GF(p) of GF(p)-rank at most r which is closest to $\boldsymbol{A}$ in the ℓ_0-*norm*, i.e., the goal is to minimize $\|\boldsymbol{A} - \boldsymbol{B}\|_0$, the number of different entries in $\boldsymbol{A}$ and $\boldsymbol{B}$. In the robust version of this problem, some of the columns of $\boldsymbol{A}$ could be outliers, which brings us to the following problem.

Robust ℓ_0-Low Rank Approximation

Input: An $m \times n$ matrix $\boldsymbol{A}$ over GF(p), a positive integer r, non-negative integers B and ℓ.

Task: Decide whether there is a matrix $\boldsymbol{B}$ of GF(p)-rank at most r, and a matrix $\boldsymbol{C}$ over GF(p) with at most ℓ non-zero columns such that $\|\boldsymbol{A} - \boldsymbol{B} - \boldsymbol{C}\|_0 \leq B$.

Note that in this definition the non-zero columns of $\boldsymbol{C}$ can take any values. However, it is easy to see that the problem would be equivalent if the columns of $\boldsymbol{C}$ were constrained to be either zero columns or the respective columns of $\boldsymbol{A}$. This holds since if $\boldsymbol{C}$ contains a non-zero column, it could be replaced by the respective column of $\boldsymbol{A}$, and the respective column of $\boldsymbol{B}$ can be replaced by a zero column. This does not increase the cost nor the rank of $\boldsymbol{B}$. Thus any of the two formulations allows to restore the column outliers in $\boldsymbol{A}$ from $\boldsymbol{C}$.

By a reduction [14, Lemma 1] similar to Lemma 5, we can show that Theorem 4 yields the following theorem.

▶ **Theorem 6.** *Robust ℓ_0-Low Rank Approximation is FPT parameterized by B when p and r are constants.*

Another popular variant of low-rank matrix approximation is the case when the approximation matrix $\boldsymbol{B}$ is of low Boolean rank. More precisely, let $\boldsymbol{A}$ be a binary $m \times n$ matrix. Now we consider the elements of $\boldsymbol{A}$ to be *Boolean* variables. The *Boolean rank* of $\boldsymbol{A}$ is the minimum r such that $\boldsymbol{A} = \boldsymbol{U} \wedge \boldsymbol{V}$ for a Boolean $m \times r$ matrix $\boldsymbol{U}$ and a Boolean $r \times n$ matrix

$\boldsymbol{V}$, where the product is Boolean, that is, the logical $\wedge$ plays the role of multiplication and $\vee$ the role of sum. The variant of the low Boolean-rank matrix approximation is the following problem.

Robust Low Boolean-Rank Approximation

Input: A binary $m \times n$ matrix $\boldsymbol{A}$, a positive integer r, non-negative integers B and ℓ.

Task: Decide whether there is a binary matrix $\boldsymbol{B}$ of Boolean rank $\leq r$, and a binary matrix $\boldsymbol{C}$ with at most ℓ non-zero columns such that $\|\boldsymbol{A} - \boldsymbol{B} - \boldsymbol{C}\|_0 \leq B$.

By Theorem 4 and reduction from constrained clustering to Boolean-rank matrix approximation identical to [14, Lemma 2], we have the following.

▶ **Theorem 7.** *Robust Low Boolean-Rank Approximation is FPT parameterized by B when r is a constant.*

Finally, we consider clustering with outliers. This problem looks very similar to feature selection. The only difference is that instead of features (the rows of the matrix $\boldsymbol{A}$), we seek to remove some columns of $\boldsymbol{A}$. More precisely, we consider the following problem.

k-Clustering with Column Outliers

Input: An alphabet Σ, an $m \times n$ matrix $\boldsymbol{A}$ with columns $\boldsymbol{a}^1, \boldsymbol{a}^2, \dots, \boldsymbol{a}^n$ such that $\boldsymbol{a}^j \in \Sigma^m$ for all $1 \leq j \leq n$, a positive integer k, non-negative integers B and ℓ.

Task: Decide whether there is a subset $O \subset \{1, 2, \dots, n\}$ of size at most ℓ, a partition of $\{1, 2, \dots, n\} \setminus O$ into k sets $\{I_1, I_2, \dots, I_k\}$ called clusters, and vectors $\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_k \in \Sigma^m$ such that the cost of clustering

$$\sum_{i=1}^{k} \sum_{j \in I_i} d_H(\boldsymbol{a}^j, \boldsymbol{c}_i) \leq B.$$

Note that k-Clustering with Column Outliers is also a special case of Constrained Clustering with Outliers when every relation $R_i \in \mathcal{R}$ contains all possible k-tuples over Σ, that is, there are no constraints on the centers. Hence, by Theorem 4, we readily obtain the same result for this problem. However, in this special case we show that it is possible to obtain an improved result.

▶ **Theorem 8.** *k-Clustering with Column Outliers is solvable in time $2^{O(B \log B)} |\Sigma|^B \cdot (nm)^{O(1)}$.*

In particular, the theorem implies that the problem is FPT parameterized by B and $|\Sigma|$. We note that the running time of Theorem 8 matches the running time in [15] obtained for the Binary k-Clustering problem without outliers on binary data. The interesting feature of the theorem is that the running time of the algorithm does not depend on the number of outliers ℓ, matching the bound of the problem without outliers. Most of the clustering procedures in robust statistics, data mining and machine learning perform well only for small number of outliers. Our theorem implies that if all of the inlier points could be naturally partitioned into k distinct clusters with small cost, then such a clustering could be efficiently recovered even after arbitrarily many outliers are added.

Related Work. Constrained Clustering (without outliers) was introduced in [14] as a tool for designing EPTAS for Low Boolean-Rank Approximation. Robust ℓ_0-Low Rank Approximation is a variant of robust PCA for categorical data. The study of robust

PCA, where one seeks for a PCA when the input data is noisy or corrupted, is the large class of extensively studied problems, see the books [30, 8]. There are many models of robustness in the literature, most relevant to our work is the approach that became popular after the work of [9]. The variant of robust PCA where one seeks for identifying a set of outliers, also known as PCA with outliers, were studied in [4, 10, 31, 29].

For the vanilla variant, ℓ_0-Low Rank Approximation, a number of parameterized and approximation algorithms were developed [3, 14, 15, 21].

Low Boolean-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. We refer to [23, 25, 27, 26] for further references on this interesting problem. Parameterized algorithms for Low Boolean-Rank Approximation (without outliers) were studied in [15].

There are several approximations and parameterized algorithms known for Binary k-Clustering, which is the vanilla (without outliers) case of k-Clustering with Column Outliers and with $\Sigma = \{0, 1\}$ [28, 14, 3, 16]. Most relevant to our work is the parameterized algorithm for Binary k-Clustering from [15]. Theorem 8 extends the result from [15] to clustering with outliers.

Paper Outline. In the remaining part of this extended abstract we focus on our algorithmic results. We briefly outline our techniques in Section 3. Then, in Section 4 we describe our main result, the FPT algorithm for Constrained Clustering with Outliers. Finally, in Section 5, we conclude with some open problems. Due to space constraints, the detailed presentation of the remaining results appears in the attached full version of the paper.

3 Our Techniques

Both of our algorithmic results, Theorems 4 and 8, have at their core the subhypergraph enumeration technique introduced by Marx [24]. This is fairly natural, since our algorithms solve generalized versions of the vanilla binary clustering problem, and the only known FPT algorithm [15] for the latter problem parameterized by B relies on the hypergraph enumeration as well. In fact, our algorithm for k-Clustering with Column Outliers closely follows this established approach of applying the hypergraph construction to clustering problems ([16], and partly [15]). However, for the Constrained Clustering with Outliers problem the existing techniques do not work immediately. To deal with this, we generalize the previously used hypergraph construction. In what follows, we present the key ideas of both algorithms. We begin with the simpler case of k-Clustering with Column Outliers, even though our main results are for Feature Selection and Constrained Clustering with Outliers.

For the presentation of our algorithms, we recall standard hypergraph notations and the notion of a fractional cover of a hypergraph. A hypergraph $G(V_G, E_G)$ consists of a set V_G of vertices and a set E_G of edges, where each edge is a subset of V_G. Consider two hypergraphs $H(V_H, E_H)$ and $G(V_G, E_G)$. We say that H appears in G at $V' \subseteq V_G$ as a partial hypergraph if there is a bijection π between V_H and V' such that for any edge $e \in E_H$, $\pi(e) \in E_G$, where $\pi(e) = \cup_{v \in e} \pi(v)$. H is said to appear in G at $V' \subseteq V_G$ as a subhypergraph if there is a bijection π between V' and V_H such that for any edge $e \in E_H$, there is an edge $e' \in E_G$ such that $\pi(e) = e' \cap V'$.

A fractional edge cover of H is an assignment $\phi : E_H \to [0,1]$ such that for every vertex $v \in V_H$, the sum of the values assigned to the edges that contain v is at least 1, i.e, $\sum_{e \ni v} \phi(e) \geq 1$. The fractional cover number $\rho^*(H)$ of H is the minimum value $\sum_{e \in E} \phi(e)$ over all fractional edge covers ϕ of H. The following theorem is required for our algorithm.

▶ **Theorem 9** ([24]). *Let $H(V_H, E_H)$ be a hypergraph with fractional cover number $\rho^*(H)$, and let $G(V_G, E_G)$ be a hypergraph where each edge has size at most L. There is an algorithm that enumerates, in time $|V_H|^{O(|V_H|)} \cdot L^{|V_H|\rho^*(H)+1} \cdot |E_G|^{\rho^*(H)+1} \cdot |V_G|^2$, every subset $V \subseteq V_G$ where H appears in G as a subhypergraph.*

3.1 The Algorithm for k-Clustering with Column Outliers

Given an instance $(\boldsymbol{A}, k, B, \ell)$ of k-Clustering with Column Outliers, we note that at most $2B$ distinct columns can belong to "nontrivial" clusters (with at least 2 distinct columns), exactly like in the case without the outliers. So we employ a color-coding scheme [2] to partition the columns in a way so that every column belonging to a nontrivial cluster of a fixed feasible solution is colored with its own color. Thus we reduce to multiple instances of the problem we call Restricted Clustering. In Restricted Clustering, we are given sets of columns $U_1, U_2, \ldots, U_p$ and a parameter B. The goal is to select p columns $\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_p}$ and a cluster center $\boldsymbol{s}$ such that $\boldsymbol{b_t} \in U_t$ for $1 \leq t \leq p$ and $\sum_{t=1}^{p} d_H(\boldsymbol{b_t}, \boldsymbol{s}) \leq B$.

Restricted Clustering is similar to the Cluster Selection problem of [16] and [15], and the hypergraph-based algorithm to solve it is essentially the same as in [16]. However, next we briefly sketch the details, as this construction serves as the base for our more general Constrained Clustering with Outliers algorithm. First, guess $\boldsymbol{b_1} \in U_1$ in the optimal solution to the instance of Restricted Clustering. If the cost of the optimal solution is at most B, then $d_H(\boldsymbol{b_1}, \boldsymbol{s})$ is at most B as well. If we know the set of at most B positions P where $\boldsymbol{b_1}$ and $\boldsymbol{s}$ differ, we can easily identify $\boldsymbol{s}$ by trying all possible $|\Sigma|^B$ options at these positions. For each option, we can find the closest $\boldsymbol{b_i}$ from each U_i and check whether the total cost is at most B.

To show that we can enumerate all potential sets of deviating positions in FPT time, we identify the instance with the following hypergraph $H(V, E)$. The vertices V are the positions $\{1, \ldots, m\}$. With each column $\boldsymbol{x}$ in $\bigcup_{i=1}^{p} U_i$, we identify a hyperedge containing exactly the positions where $\boldsymbol{x}$ is different from $\boldsymbol{b_1}$. Now the optimal set of positions P and the optimal columns $\{\boldsymbol{b_i}\}_{i=1}^{p}$ induce a subhypergraph $H_0(V_0, E_0)$ of H such that $|V_0| \leq B$ and the fractional cover number of H_0 is at most two. The latter holds simply because wherever $\boldsymbol{s}$ is different from $\boldsymbol{x}$, at least half of the chosen columns must also be different from $\boldsymbol{x}$, otherwise modifying $\boldsymbol{s}$ to match $\boldsymbol{x}$ decreases the cost. If we enumerate all possible subhypergraphs H_0 and all possible locations in H where they occur, we can surely find the optimal set of locations P. Since $|V_0| \leq B$, enumerating all choices for H_0 is clearly in FPT time. For a fixed H_0, finding all occurrences in H is in FPT time by Theorem 9. Note that applying Theorem 9 results in FPT time only when the fractional cover number of H_0 is bounded by a constant. Also, by a sampling argument one can show that it suffices to consider only those H_0 with $O(\log B)$ hyperedges. It follows that the number of distinct hypergraphs that we have to consider for enumeration is bounded by only $2^{O(B \log B)}$. Thus it is possible to bound the dependence on B in the running time by $2^{O(B \log B)}$.

3.2 The Algorithm for Feature Selection

For Feature Selection, the above-mentioned approach is not applicable, for several reasons. Most crucially, it does not seem that one can partition the problem into k independent instances of a simpler single-center selection problem, in a way that we reduce k-Clustering with Column Outliers to k instances of Restricted Clustering (for a fixed coloring). Intuitively, the possibility to remove a subset of features does not allow such a partition as all the clusters depend simultaneously on the choice of those features. Moreover, our hardness result shows that for Feature Selection the running time cannot match with k-Clustering with Column Outliers, as no $n^{o(k)}$ time algorithm is possible for constant B, assuming ETH.

By Lemma 5, for solving Feature Selection, it suffices to solve Constrained Clustering with Outliers. For the same reasons as with Feature Selection, the approach used for k-Clustering with Column Outliers fails, as the constraints on the centers do not allow to form clusters independently. Instead, we generalize the hypergraph construction used for Restricted Clustering to handle the choice of all k centers simultaneously, as opposed to just one center at a time. This is the most technical part of the paper. The main idea is to base the hypergraph on k-tuples of columns instead of just single columns. In the next section, we formalize this intuitive discussion, presenting the proof in full detail.

4 The Algorithm for Constrained Clustering with Outliers

In this section we prove Theorem 4 by giving an algorithm for Constrained Clustering with Outliers that runs in $(kB)^{O(kB)}n^{O(k)}m^2|\Sigma|^{kB}$ time. First, we prove a structural lemma that will be useful for analysis of the algorithm.

4.1 Structural Lemma

Here we show that an optimal set of centers corresponds to a "good" subhypergraph in a certain hypergraph. Consider a feasible clustering $\mathcal{I} = \{I_1, I_2, \ldots, I_k\}$ having the minimum cost. Let $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ be a fixed set of centers corresponding to $\mathcal{I}$. Also, let T be the set of all tuples of the form $(\boldsymbol{a}^{i_1}, \boldsymbol{a}^{i_2}, \ldots, \boldsymbol{a}^{i_k})$ such that $i_j \in I_j$ for all j. Note that we do not actually need to know the set T – we just introduce the notation for the sake of analysis. For a k-tuple $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$, we denote the tuple $(\boldsymbol{x}_1[j], \boldsymbol{x}_2[j], \ldots, \boldsymbol{x}_k[j])$ by $x[j]$. Two k-tuples x and y are said to differ from each other at location j if $x[j] \neq y[j]$.

Let C be the k-tuple such that $C = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k)$. Suppose $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ is such that there are at most B positions h where $x[h] \neq C[h]$, and for each $1 \leq j \leq m$, $x[j] \in R_j$. Consider the hypergraph H defined in the following way with respect to x. The labels of the vertices of H are in $\{1, 2, \ldots, m\}$. For each k-tuple $y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ of T, we add an edge $S \subseteq \{1, 2, \ldots, m\}$ such that $h \in S$ if $x[h] \neq y[h]$.

In the following lemma, we show that the hypergraph H has a "good" subhypergraph.

▶ **Lemma 10** (Structural Lemma). *Suppose the input is a yes-instance. Consider a k-tuple $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ such that there are at most B positions h where $x[h] \neq C[h]$ and for each $1 \leq j \leq m$, $x[j] \in R_j$. Also, consider the hypergraph H defined in the above with respect to x. There exists a subhypergraph $H^*(V^*, E^*)$ of H with the following properties.*

1. $|V^*| \leq B$.
2. $|E^*| \leq 200 \ln B$.
3. *The indices in V^* are the exact positions h such that $x[h] \neq C[h]$.*
4. *The fractional cover number of H^* is at most 4.*

To prove the above lemma, first, we show the existence of a subhypergraph that satisfies all the properties except the second one. Let P be the set of positions h such that $x[h] \neq C[h]$. Let $H_0(V_0, E_0)$ be the subhypergraph of H induced by P. By definition of x, P contains at most B indices. Thus, the first property follows immediately. The third property also follows, as $V_0 = P$, is exactly the set of positions $h \in \{1, 2, \ldots, m\}$, where $x[h] \neq C[h]$. Next, we show that the fourth property holds for H_0. In fact, we show a stronger bound.

▶ **Lemma 11.** *The fractional cover number of H_0 is at most 2.*

Proof. Note that the total number of edges of H_0 is $\tau = |T|$. We claim that each vertex of H_0 is contained in at least $\tau/2$ edges.

Consider any vertex h of H_0. Suppose there is a $1 \leq j \leq k$, such that for at least $\lceil |I_j|/2 \rceil$ columns in I_j the value at location h is not $\boldsymbol{x}_j[h]$. Note that each such column contributes to $\Pi_{t^1=1}^{j-1}|I_{t^1}| \cdot \Pi_{t^2=j+1}^{k}|I_{t^2}| = \tau/|I_j|$ tuples $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ of T such that $\boldsymbol{y}_j[h] \neq \boldsymbol{x}_j[h]$. Thus, the edge corresponding to each such tuple contains h. It follows that, at least $\lceil |I_j|/2 \rceil \cdot (\tau/|I_j|) \geq \tau/2$ edges in E_0 contain h.

In the other case, for all $1 \leq j \leq k$ and less than $\lceil |I_j|/2 \rceil$ columns in I_j, the value at location h is not $\boldsymbol{x}_j[h]$. We prove that this case does not occur. Note that there is a k-tuple z in R_h such that $z = x[h]$. Consider replacing $C[h]$ by z at position h of C. Next, we analyze the change in cost of the clustering $\mathcal{I}$. Note that the cost corresponding to positions other than h remains same. For a $1 \leq j \leq k$, if previously $\boldsymbol{c}_j[h] = \boldsymbol{x}_j[h]$, the cost remains same. Otherwise, $\boldsymbol{c}_j[h] \neq \boldsymbol{x}_j[h]$. Note that for more than $\lceil |I_j|/2 \rceil$ columns in I_j, the value at location h is $\boldsymbol{x}_j[h]$. Thus, by replacing $\boldsymbol{c}_j[h]$ by $\boldsymbol{x}_j[h]$, the cost decrement corresponding to the index j and location h is at least 1. As $x[h] \neq C[h]$, there is an index j such that $\boldsymbol{c}_j[h] \neq \boldsymbol{x}_j[h]$. It follows that the overall cost decrement is at least 1, which contradicts the optimality of the previously chosen centers. Hence, this case cannot occur. This completes the proof of the lemma. ◀

So far we have proved the existence of a subhypergraph that satisfies all the properties except the second. Next, we show the existence of a subhypergraph that satisfies all the properties. The following lemma completes the proof of Lemma 10. Its proof follows a standard sampling argument, and is presented in the full version.

▶ **Lemma 12.** *Let $B \geq 2$. Consider the subhypergraph H_0 that satisfies all the properties of Lemma 10 except (2). It is possible to select at most $200 \ln B$ edges from H_0 such that the subhypergraph H_0^* obtained by removing all the other edges from H_0 satisfies all the properties of Lemma 10.*

4.2 The Algorithm for Constrained Clustering

In this section, we describe our algorithm. The algorithm outputs a feasible clustering of minimum cost if there is a feasible clustering of the given instance. Otherwise, the algorithm returns "NO".

The Algorithm. First, we consider all k-tuples $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ such that $\boldsymbol{x}_j$ is a column of $\boldsymbol{A}$ for $1 \leq j \leq k$, and apply the following refinement process on each of them. Here, a k-tuple x of columns of $\boldsymbol{A}$ is said to differ from $\mathcal{R}$ at a position j for $1 \leq j \leq m$ if $x[j] \notin R_j$.

- Let $P \subseteq \{1, 2, \ldots, m\}$ be the set of positions where x differs from $\mathcal{R}$.
- If $|P| > B$, probe the next k-tuple x.
- For each position $h \in P$, replace $x[h]$ by any element of R_h.

Next, for each refined k-tuple $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$, we construct a hypergraph G whose vertices are in $\{1, 2, \ldots, m\}$. For each k-tuple $y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ of columns of $\boldsymbol{A}$, we add an edge $S \subseteq \{1, 2, \ldots, m\}$ such that $h \in S$ if $x[h] \neq y[h]$. For all hypergraphs H_0^* having at most B vertices and at most $200 \ln B$ edges, we check if each vertex of H_0^* is contained in at least $1/4$ fraction of the edges. If that is the case, we use the algorithm of Theorem 9 to find every place P' where H_0^* appears in G as subhypergraph. For each such set P', we perform all possible $B' \leq B \cdot k$ edits of the tuple x at the locations in P'. In particular, for each B', the editing is done in the following way. For each possible B' entries $(a_1, \ldots, a_{B'})$ in $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ at the locations in P' and each set of B' symbols $(s_1, \ldots, s_{B'})$ from Σ, we put s_j at the entry a_j for all j. After each such edit, we retrieve the edited tuple $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ and perform a sanity check on this tuple to ensure that it is a valid k-tuple center. In particular, for each index $1 \leq h \leq m$, if there is a $z \in R_h$ such that $z = x[h]$, we tag x as a valid tuple. Lastly, we output all such valid k-tuples as candidate centers if the corresponding cost of clustering is at most B. If no k-tuple is output as a candidate center, we return "NO".

Note that, given a k-tuple candidate center $z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k)$, one can compute a minimum cost clustering in the following greedy way, which implies that we can correctly compute the cost of clustering in the above. At each step i, we assign a new column of $\boldsymbol{A}$ to a center. In particular, we add a column $\boldsymbol{a}^j$ of $\boldsymbol{A}$ to a cluster I'_t such that $\boldsymbol{a}^j$ incurs the minimum cost over all unassigned columns if it is assigned to a center, i.e, it minimizes the quantity $\min_{t'=1}^{k} d(\boldsymbol{a}^j, \boldsymbol{z}_{t'})$, and $\boldsymbol{z}_t$ is a corresponding center nearest to $\boldsymbol{a}^j$. As we are allowed to exclude ℓ outliers, we assign $n - \ell$ columns. The clustering $\{I'_1, \ldots, I'_k\}$ obtained at the end of this process is the output. This finishes the description of our algorithm.

4.3 Analysis

Again consider the feasible clustering with partition $\mathcal{I} = \{I_1, I_2, \ldots, I_k\}$ and the corresponding center tuple $C = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k)$ having the minimum cost. First, we have the following observation.

▶ **Observation 13.** *Suppose for a k-tuple x, x differs from C at B_1 positions. Then, after refinement, there is at most B_1 positions h such that $x[h] \neq C[h]$. Moreover, after refinement, $d_H(x, C) \leq B_1 \cdot k$.*

The first part is true for the following reason. If x was different from $\mathcal{R}$ at a position h, then $x[h] \neq C[h]$ as well. Thus, refinement is applied for a position h where $x[h]$ already differs from $C[h]$. Hence, refinement does not affect a position h where $x[h] = C[h]$. The moreover part follows trivially from the first part as x is a k-tuple. Now, it is sufficient to prove the following lemma.

▶ **Lemma 14.** *Suppose there is a feasible clustering with partition $\mathcal{I} = \{I_1, I_2, \ldots, I_k\}$ as defined above. The above algorithm successfully outputs the k-tuple $(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k)$.*

Proof. Consider a k-tuple $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ such that the column $\boldsymbol{x}_j$ is in cluster I_j. As the algorithm considers all possible k-tuples of columns in $\boldsymbol{A}$, it must consider x. By Observation 13, after refinement, there are at most B positions h where $x[h] \neq C[h]$. Also, for each $1 \leq j \leq m$, $x[j] \in R_j$. Let G be the hypergraph constructed by the algorithm corresponding to this refined x. Note that the hypergraph H defined in Lemma 10 is a partial subhypergraph of G. Thus, the subhypergraph H^* of H is also a subhypergraph of G. As we enumerate all hypergraphs having at most B vertices, at most $200 \ln B$ edges and at most 4 fractional covering number, H^* must be considered by the algorithm. Let P' be the place in G where H^* appears. By the third property of Lemma 10, the locations in P' are the exact positions h such that $x[h] \neq C[h]$. It follows that an edit corresponding to P' generates the tuple $C = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k)$, as $d_H(x, C) \leq B \cdot k$. It is easy to see that C also passes the sanity check. Hence, C must be an output of the algorithm. ◀

Given the tuple center $C = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k)$, we use the greedy assignment scheme (described in the algorithm) to find the underlying clustering. Note that given any k-tuple candidate center $z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k)$, this greedy scheme computes a minimum cost clustering with

$\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ being the cluster centers. Thus, the cost of the clustering computed by the algorithm is at most B. Hence, the algorithm successfully outputs C as a candidate center. We summarize our findings in the following lemma.

▶ **Lemma 15.** *Suppose the input instance is a no-instance, then the algorithm successfully outputs "NO". If the input instance is a yes-instance, the algorithm correctly computes a feasible clustering.*

4.4 Time Complexity

Next, we discuss the time complexity of our algorithm. The number of choices of x is $n^{O(k)}$. For each choice of x, the hypergraph G can be constructed in $n^{O(k)}$ time. The number of distinct hypergraphs H_0^* with at most B vertices and at most $200 \ln B$ edges is $2^{B \cdot 200 \ln B} = B^{O(B)}$, since there are 2^B possibilities for each edge. Now we analyze the time needed for locating a particular H_0^* in G. For any tuple $y \in T$, $d_H(y, C) \leq B$. By triangle inequality, $d_H(x, y) \leq 2B$. Thus, the size of any edge in H is at most $2B$, and we can remove any edge of G of size more than $2B$. From Theorem 9, it follows that every occurrence of H_0^* in G can be found in $B^{O(B)} \cdot (2B)^{4B+1} \cdot n^{4k+k} \cdot m^2 = B^{O(B)} \cdot n^{O(k)} m^2$ time. If H_0^* appears at some place in G, it would take $O((kB|\Sigma|)^{kB})$ time to edit x. Hence, in total the algorithm takes $(kB)^{O(kB)} |\Sigma|^B \cdot n^{O(k)} m^2$ time. By the above discussion, we have Theorem 4.

5 Conclusion

We initiated the systematic study of parameterized complexity of robust categorical data clustering problems. In particular, for k-Clustering with Column Outliers, we proved that the problem can be solved in $2^{\mathcal{O}(B \log B)} |\Sigma|^B \cdot (nm)^{\mathcal{O}(1)}$ time. Further, we considered the case of row outliers and proved that Feature Selection is solvable in time $f(k, B, |\Sigma|) \cdot m^{g(k,|\Sigma|)} n^2$. We also proved that we cannot avoid the dependence on k in the degree of the polynomial of the input size in the running time unless W[1] = FPT, and the problem cannot be solved in $m^{o(k)} \cdot n^{\mathcal{O}(1)}$ time, unless ETH is false. To deal with row outliers, we introduced the Constrained Clustering with Outliers problem and obtained the algorithm with running time $(kB)^{\mathcal{O}(kB)} |\Sigma|^{kB} \cdot m^2 n^{\mathcal{O}(k)}$. This problem is very general, and the algorithm for it not only allowed us to get the result for Feature Selection, but also led to the algorithms for the robust low rank approximation problems. In particular, we obtained that Robust ℓ_0-Low Rank Approximation is FPT if k and p are constants when the problem is parameterized by B. However, even if the low rank approximation problems are closely related to the matrix clustering problems, there are structural differences. For instance, we show that the complexity of clustering with column outliers is different from row outliers, however, low-rank approximation problems are symmetric. This leads to the question whether Robust ℓ_0-Low Rank Approximation, Robust Low Boolean-Rank Approximation and Robust Projective Clustering could be solved by better algorithms specially tailored for these problems. It is unlikely that potential improvements would considerably change the general qualitative picture. For example, Robust ℓ_0-Low Rank Approximation for $p = 2$ and $\ell = 0$ is NP-complete if $k = 2$ [12, 19] and W[1]-hard when parameterized by B [17]. It is also easy to observe that Robust ℓ_0-Low Rank Approximation for $p = 2$, $B = 0$ and $k = n - \ell - 1$ is equivalent to asking whether the input matrix $\boldsymbol{A}$ has $n - \ell$ linearly dependent columns. This immediately implies that Robust ℓ_0-Low Rank Approximation for $p = 2$ and $B = 0$ is W[1]-hard when parameterized by k or $n - \ell$ by the recent results about the Even Set problem [5]. The most interesting open question, by our opinion, is whether the exponential dependence on k in the degree of the polynomial of the input size in the running time produced by our reduction of Robust ℓ_0-Low Rank Approximation to Constrained Clustering with Outliers could be avoided, even if p is a constant. Can the dependence of k be made polynomial (or even linear)?

References

1 Salem Alelyani, Jiliang Tang, and Huan Liu. Feature selection for clustering: A review. In Charu C. Aggarwal and Chandan K. Reddy, editors, *Data Clustering: Algorithms and Applications*, pages 30–373. CRC Press, 2013.

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

3 Frank Ban, Vijay Bhattiprolu, Karl Bringmann, Pavel Kolev, Euiwoong Lee, and David P. Woodruff. A PTAS for ℓ_p-low rank approximation. In *SODA'19*, pages 747–766. SIAM, 2019. doi:10.1137/1.9781611975482.47.

4 Aditya Bhaskara and Srivatsan Kumar. Low Rank Approximation in the Presence of Outliers. In *APPROX/RANDOM'18*, volume 116, pages 4:1–4:16, Dagstuhl, Germany, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.4.

5 Arnab Bhattacharyya, Édouard Bonnet, László Egri, Suprovat Ghoshal, Karthik C. S., Bingkai Lin, Pasin Manurangsi, and Dániel Marx. Parameterized intractability of even set and shortest vector problem. *CoRR*, abs/1909.01986, 2019. arXiv:1909.01986.

6 Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. Unsupervised feature selection for the k-means clustering problem. In *NIPS'09*, pages 153–161. Curran Associates, Inc., 2009. URL: http://papers.nips.cc/paper/3724-unsupervised-feature-selection-for-the-k-means-clustering-problem.

7 Christos Boutsidis, Anastasios Zouzias, Michael W. Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *IEEE Trans. Information Theory*, 61(2):1045–1062, 2015. doi:10.1109/TIT.2014.2375327.

8 Thierry Bouwmans, Necdet Serhat Aybat, and El-hadi Zahzah. *Handbook of robust low-rank and sparse matrix decomposition: Applications in image and video processing*. Chapman and Hall/CRC, 2016.

9 Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, 2011. doi:10.1145/1970392.1970395.

10 Yudong Chen, Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust matrix completion and corrupted columns. In *ICML'11*, pages 873–880, 2011.

11 Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *STOC'15*, pages 163–172. ACM, 2015.

12 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.

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

14 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Approximation schemes for low-rank binary matrix approximation problems. *ACM Trans. Algorithms*, 16(1):12:1–12:39, 2020. doi:10.1145/3365653.

15 Fedor V. Fomin, Petr A. Golovach, and Fahad Panolan. Parameterized low-rank binary matrix approximation. *Data Min. Knowl. Discov.*, 34(2):478–532, 2020. doi:10.1007/s10618-019-00669-5.

16 Fedor V. Fomin, Petr A. Golovach, and Kirill Simonov. Parameterized k-Clustering: Tractability Island. In *FSTTCS'19*, volume 150 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.FSTTCS.2019.14.

17 Fedor V. Fomin, Daniel Lokshtanov, Syed Mohammad Meesum, Saket Saurabh, and Meirav Zehavi. Matrix rigidity from the viewpoint of parameterized complexity. *SIAM J. Discrete Math.*, 32(2):966–985, 2018. doi:10.1137/17M112258X.

18 Robert Ganian, Iyad Kanj, Sebastian Ordyniak, and Stefan Szeider. On the parameterized complexity of clustering incomplete data into subspaces of small rank. In *AAAI'20*, pages 3906–3913. AAAI Press, 2020.

19 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. arXiv:1509.09236.

20 YongSeog Kim, W. Nick Street, and Filippo Menczer. Evolutionary model selection in unsupervised learning. *Intell. Data Anal.*, 6(6):531–556, 2002. URL: http://content.iospress.com/articles/intelligent-data-analysis/ida00110.

21 Ravi Kumar, Rina Panigrahy, Ali Rahimi, and David P. Woodruff. Faster algorithms for binary matrix factorization. In *ICML'19*, volume 97 of *Proceedings of Machine Learning Research*, pages 3551–3559. PMLR, 2019. URL: http://proceedings.mlr.press/v97/kumar19a.html.

22 Tao Li. A general model for clustering binary data. In *KDD'05*, pages 188–197, 2005.

23 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.

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

25 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.

26 Pauli Miettinen and Stefan Neumann. Recent developments in boolean matrix factorization. In *IJCAI'20*, pages 4922–4928. ijcai.org, 2020.

27 Pauli Miettinen and Jilles Vreeken. Model order selection for boolean matrix factorization. In *KDD'11*, pages 51–59. ACM, 2011. doi:10.1145/2020408.2020424.

28 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.

29 Kirill Simonov, Fedor V. Fomin, Petr A. Golovach, and Fahad Panolan. Refined complexity of PCA with outliers. In *ICML'19*, volume 97, pages 5818–5826. PMLR, 2019. URL: http://proceedings.mlr.press/v97/simonov19a.html.

30 René Vidal, Yi Ma, and S. Shankar Sastry. *Generalized Principal Component Analysis*, volume 40 of *Interdisciplinary applied mathematics*. Springer, 2016. doi:10.1007/978-0-387-87811-9.

31 Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust PCA via outlier pursuit. In *NIPS'10*, pages 2496–2504. Curran Associates, Inc., 2010. URL: http://papers.nips.cc/paper/4005-robust-pca-via-outlier-pursuit.

32 Zhongyuan Zhang, Tao Li, Chris Ding, and Xiangsun Zhang. Binary matrix factorization with applications. In *ICDM'07*, pages 391–400. IEEE, 2007.