12-1-2023

Parameterized Complexity of Feature Selection for Categorical Data Clustering

Sayan Bandyapadhyay
Portland State University, sayanb@pdx.edu

Fedor V. Fomin
University of Bergen

Petr A. Golovach
University of Bergen

Kirill Simonov
University of Potsdam

Citation Details

Bandyapadhyay, S., Fomin, F. V., Golovach, P. A., & Simonov, K. (2023). Parameterized complexity of feature selection for categorical data clustering. ACM Transactions on Computation Theory, 15(3-4), 1-24.

Parameterized Complexity of Feature Selection for Categorical Data Clustering

SAYAN BANDYAPADHYAY, Department of Computer Science, Portland State University, USA
FEDOR V. FOMIN and PETR A. GOLOVACH, Department of Informatics, University of Bergen, Norway
KIRILL SIMONOV, Hasso Plattner Institute, University of Potsdam, Germany

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 and Binary and Boolean Low-rank Matrix Approximation with Outliers. Thus, as a byproduct of our theorem, we obtain algorithms for all these problems. We also complement our algorithmic findings with complexity lower bounds.

CCS Concepts: • **Mathematics of computing → Combinatorial algorithms**; • **Theory of computation → Parameterized complexity and exact algorithms;**

Additional Key Words and Phrases: Robust clustering, PCA, Low-rank approximation, hypergraph enumeration

A preliminary version of this article appeared as an extended abstract in the proceedings of MFCS 2021. This work is supported by the Research Council of Norway via the projects BWCA (grant no. 314528) and DFG Research Group ADYN via grant DFG 411362735.

Authors' addresses: S. Bandyapadhyay, Computer Science Department, Portland State University, PO Box 751, Portland, OR 97207-0751; e-mail: sayanb@pdx.edu; F. V. Fomin and P. A. Golovach, Department of Informatics, University of Bergen, P.O. Box 7803, N-5020 Bergen, Norway; e-mails: {fedor.fomin, petr.golovach}@uib.no; K. Simonov, Hasso-Plattner-Institut für Digital Engineering gGmbH, Postfach 900460, D-14440 Potsdam; e-mail: kirillsimonov@gmail.com.

1942-3454/2023/12-ART7 $15.00
https://doi.org/10.1145/3604797

ACM Reference format:
Sayan Bandyapadhyay, Fedor V. Fomin, Petr A. Golovach, and Kirill Simonov. 2023. Parameterized Complexity of Feature Selection for Categorical Data Clustering. *ACM Trans. Comput. Theory* 15, 3-4, Article 7 (December 2023), 24 pages.
https://doi.org/10.1145/3604797

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 of 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 [7, 8, 12].

Kim et al. [24] 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 market survey data that typically contains responses of customers to a set of questions regarding their demographic and psychographic information, shopping experience, attitude toward 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. [24] 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 clusters 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, and so forth. 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 [26, 36]. 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 article, we introduce a new model of feature selection w.r.t. categorical data clustering, which is motivated by the work of Kim et al. [24]. 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, $d_H(\boldsymbol{x}, \boldsymbol{y}) = |\{i \in \{1, \ldots, m\} : \boldsymbol{x}[i] \neq \boldsymbol{y}[i]\}|$. For a set of indices $S \subset \{1, 2, \ldots, m\}$ 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 $\boldsymbol{a}^1, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^n$ 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 $O \subset \{1, 2, \ldots, m\}$ of size at most ℓ, a partition $\{I_1, I_2, \ldots, I_k\}$ of $\{1, 2, \ldots, n\}$, and vectors $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \in \Sigma^{m-|O|}$ 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 $\ell = 0$ and $\Sigma = \{0, 1\}$, Feature Selection is the popular Binary k-Clustering problem, which is known to be NP-hard for every $k \geq 2$ [16]. This makes it natural to investigate the parameterized complexity of Feature Selection.

Our main result is the following theorem.

Theorem 2.1. *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 Theorem 2.1, $f(k, B, |\Sigma|) = (k'B)^{O(k'B)}|\Sigma|^{k'B}$ and $g(k, |\Sigma|) = O(k')$, where $k' = |\Sigma|^k$. 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 2.1 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 [21] on matrix completion and [18] 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, $|\Sigma| = 2$.

[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.

Another interesting property of our algorithm is that the running time does not depend on the number ℓ of irrelevant features. 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 ℓ of irrelevant features 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 number of clusters k. In this regard, one can compare our result with the result in [18] 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 2.2. *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* $f(k) \cdot m^{o(k)} \cdot n^{O(1)}$ *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 2.1 it follows that Feature Selection can be solved in time $f(k) \cdot m^{g(k)} \cdot n^2$. Theorem 2.2 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 2.1, 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 [17] 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. 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 2.3 (**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 $1 \leq i \leq m$, the p-tuple formed by the ith 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 $1 \leq j \leq n$, 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

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

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 2.4. *Constrained Clustering with Outliers is solvable in* $(kB)^{O(kB)}|\Sigma|^{kB} \cdot n^{O(k)} \cdot m^2$ *time.*

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

Lemma 2.5. *For any instance* $(\boldsymbol{D}, k, B, \ell)$ *of Feature Selection, one can construct in time* $O(mn + k \cdot |\Sigma|^k)$ *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 2.1 follows from Theorem 2.4 and Lemma 2.5. Connections of constrained clustering with several other clustering and low-rank matrix approximation problems have been established in the literature [17]. Similarly, Theorem 2.4 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 for a positive integer p), the task is to find an $m \times n$ matrix $\boldsymbol{B}$ over GF(p) of GF(p)-rank at most r that 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 or 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 [17, Lemma 1] similar to Lemma 2.5, we can show that Theorem 2.4 yields the following theorem.

Theorem 2.6. *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 2.4 and a reduction (identical to [17, Lemma 2]) from Robust Low Boolean-Rank Approximation to Constrained Clustering with Outliers, we have the following.

Theorem 2.7. *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 $\boldsymbol{a}^1, \boldsymbol{a}^2, \ldots, \boldsymbol{a}^n$ 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 $O \subset \{1, 2, \ldots, n\}$ of size at most ℓ, a partition of $\{1, 2, \ldots, n\} \setminus O$ into k sets $\{I_1, I_2, \ldots, I_k\}$ called clusters, and vectors $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \in \Sigma^m$ 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 2.4, 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 2.8. *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 2.8 matches the running time in [18] 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 a small number of outliers. Our theorem

implies that if all of the inlier points could be naturally partitioned into k distinct clusters of 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 [17] 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 a large class of extensively studied problems; see [9, 34]. There are many models of robustness in the literature; most relevant to our work is the approach that became popular after the work of [10]. The variant of robust PCA where one seeks for identifying a set of outliers, also known as PCA with outliers, was studied in [5, 11, 33, 35].

For the vanilla variant, ℓ_0-Low Rank Approximation, a number of parameterized and approximation algorithms were developed [4, 17, 18, 25].

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 [27, 29–31] for further references on this interesting problem. Parameterized algorithms for Low Boolean-Rank Approximation (without outliers) were studied in [18].

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 $\Sigma = \{0, 1\}$ [4, 17, 19, 32]. Most relevant to our work is the parameterized algorithm for Binary k-Clustering from [18]. Theorem 2.8 extends the result from [18] to clustering with outliers.

Article Outline. The remaining part of this work is organized as follows. We briefly outline our techniques in Section 3. In Section 4 we describe the FPT algorithm for Constrained Clustering with Outliers. The improved FPT algorithm for k-Clustering with Column Outliers follows in Section 5. Next, in Section 6 we present the reduction from Feature Selection to Constrained Clustering with Outliers. The hardness results are deferred to Section 7. Finally, in Section 8, we conclude with some open problems.

3 OUR TECHNIQUES

In this section, we give a high-level overview of the techniques used to show our main algorithmic results; the details and formal proofs are given in Sections 4–6.

Both of our algorithmic results from Theorems 2.4 and 2.8 have at their core the subhypergraph enumeration technique introduced by Marx [28]. This is fairly natural since our algorithms solve generalized versions of the vanilla binary clustering problem, and the only known FPT algorithm [18] for the latter problem parameterized by B relies on 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 ([19], and partly [18]). 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 $\phi : E_H \rightarrow [0, 1]$ 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 3.1 ([28]). *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 two 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 [19] and [18], and the hypergraph-based algorithm to solve it is essentially the same as in [19]. 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 P of at most B positions 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 $H(V, E)$ for which we apply the crucial structural result from Lemma 4.1. 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 P of positions 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 $\boldsymbol{b}_1$, at least half of the chosen columns must also be different from $\boldsymbol{b}_1$; otherwise modifying $\boldsymbol{s}$ to match $\boldsymbol{b}_1$ 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 3.1. Note that applying Theorem 3.1 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 $2^{O(B \log B)}$.

A specific issue we need to deal with in the k-Clustering with Column Outliers problem is the following. If we encounter a number of identical columns, we need to treat them as an indivisible set (*initial cluster*), as only a cluster with sufficiently many different columns can have large cost. However, a potential flaw of this approach is that in an optimal solution, identical columns can belong to different clusters and to the set of outliers. To handle this, we prove in Lemma 5.1 that the optimal solution can always be "normalized." Namely, we can rearrange clusters of the optimal solution in a way that at most one initial cluster is split between a solution cluster and the set of outliers, and every other initial cluster is completely contained in either one of the resulting clusters or the set of outliers. Then, guessing the one initial cluster being split solves the issue. After the color-coding and solving Restricted Clustering, each remaining initial cluster either is a trivial cluster in the solution or belongs to the outliers. This can be decided greedily.

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 the 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 2.5, 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 article. The main idea is to base the hypergraph on k-tuples of columns instead of just single columns. In the next sections, 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 2.4 by giving an algorithm for Constrained Clustering with Outliers that runs in $(kB)^{O(kB)}|\Sigma|^{kB} \cdot n^{O(k)} \cdot m^2$ time. First, we prove a structural lemma that will be useful for the 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 4.1 (Structural Lemma). *Suppose the input is a yes-instance. Consider a k-tuple $x = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ (not necessarily in T) 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 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 4.2. *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 set the fractional weight of each edge to $2/\tau$, making the total weight of the edges 2. We claim that each vertex of H_0 is contained in at least $\tau/2$ edges. This proves that each vertex is covered by a set of edges with the total weight at least 1.

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.

We show that the other case does not occur, completing the proof of the lemma. 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 the cost of the clustering $\mathcal{I}$. Note that the cost corresponding to positions other than h remains the same. For a $1 \leq j \leq k$, if previously $\boldsymbol{c}_j[h] = \boldsymbol{x}_j[h]$, the cost remains the 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. □

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 4.1.

Lemma 4.3. *Let $B \geq 2$. Consider a subhypergraph H_0 that satisfies all the properties of Lemma 4.1 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 4.1.*

For the proof of the lemma, we need the following form of Chernoff bound.

Theorem 4.4 ([3]). *Let $X_1, \ldots, X_n$ be independent 0-1 random variables with $Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$. Then for $0 < \beta \leq 1$,*

$$Pr[X \leq (1-\beta)\mu] \leq exp(-\beta^2\mu/2),$$
$$Pr[X \geq (1+\beta)\mu] \leq exp(-\beta^2\mu/3).$$

Now, the proof of Lemma 4.3 is as follows.

Proof. Recall that τ is the number of edges in H_0. Construct a hypergraph H_0' by selecting each edge of H_0 independently at random with probability $\frac{150 \ln B}{\tau}$. The expected number of edges selected is $150 \ln B$. Also, by Theorem 4.4, with $\beta = 1/3$, the probability that at least $200 \ln B$ edges are selected is at most $exp(-150 \ln B/27) < 1/B^2$. Each vertex of H_0 is contained in at least $\tau/2$ edges. Thus, the expected number of edges that cover a vertex of H_0' is at least $75 \ln B$. Moreover, by Theorem 4.4, with $\beta = 1/3$, the probability that a given vertex of H_0^* is covered by at most $50 \ln B$ edges is at most $exp(-75 \ln B/18) < 1/B^3$. Therefore, with probability at least $1 - 1/B^2 - B \cdot 1/B^3$, H_0' contains at most $200 \ln B$ edges and each vertex of H_0' is covered by at least $50 \ln B$ edges. Thus, the fractional cover number of H_0' is at most 4. Hence, the second and the fourth property are satisfied. As the vertex set does not get changed, the first and the third property also hold. It follows that there exists a subhypergraph H_0^* that contains at most $200 \ln B$ edges from H_0 and satisfies all the properties of Lemma 4.1. □

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 3.1 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 valid 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 4.5. *Suppose for a k-tuple x, x differs from C at B_1 positions. Then, after refinement, there are 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 4.6. *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 4.5, 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 4.1 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 4.1, 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 4.7. *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 $y \in T$, y differs from C in at most B positions. Hence, x differs from y in at most $2B$ positions. 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 3.1, 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 $(kB)^{O(kB)}|\Sigma|^B \cdot n^{O(k)} \cdot m^2$ time. By the above discussion, we have Theorem 2.4.

Theorem 2.4. *Constrained Clustering with Outliers is solvable in* $(kB)^{O(kB)}|\Sigma|^{kB} \cdot n^{O(k)} \cdot m^2$ *time.*

5 FPT ALGORITHM FOR k-CLUSTERING WITH COLUMN OUTLIERS PARAMETERIZED BY B

In this section, we give a proof of Theorem 2.8. Let us recall that the theorem states that *k-Clustering with Column Outliers is solvable in time* $2^{O(B \log B)}|\Sigma|^B \cdot (nm)^{O(1)}$.

Let $(\boldsymbol{A}, k, B, \ell)$ be an input of k-Clustering with Column Outliers, where $\boldsymbol{A}$ is an $m \times n$ matrix, k the number of clusters, B the cost of the solution, and ℓ the number of outliers.

Let β be the number of pairwise distinct columns in $\boldsymbol{A}$. We start with partitioning the columns of $\boldsymbol{A}$ into sets of identical columns $\mathcal{J} = \{J_1, J_2, \ldots, J_\beta\}$. That is, all columns in each J_i are equal. We refer to such a set J_i as an *initial cluster*.

We show that for any yes-instance, there is a feasible solution such that the columns of at most one initial cluster is "split" by the solution. More formally, consider a clustering $C = \{I_1, I_2, \ldots, I_k\}$ of $\{1, 2, \ldots, n\} \setminus O$. Then each of the initial clusters J_i is exactly one of the following types *with respect to* C:

(i) $\{J_i\}$ such that there is t with $J_i \subseteq I_t$;
(ii) $\{J_i\}$ such that J_i is not of type (i), and $J_i \subseteq \cup_{j=1}^{k} I_j$;
(iii) $\{J_i\}$ such that $J_i \subseteq O$;
(iv) $\{J_i\}$ such that $J_i \not\subseteq \cup_{j=1}^{k} I_j$, $J_i \not\subseteq O$, and there is t with $J_i \subseteq I_t \cup O$; and
(v) $\{J_i\}$ such that $J_i \not\subseteq \cup_{j=1}^{k} I_j$, $J_i \not\subseteq O$, and there is no t with $J_i \subseteq I_t \cup O$.

Lemma 5.1. *Let $(\boldsymbol{A}, k, B, \ell)$ be a yes-instance of k-Clustering with Column Outliers. Then there is a set $O \subseteq \{1, 2, \ldots, n\}$ of size ℓ and k-clustering C of $\{1, 2, \ldots, n\} \setminus O$ of cost at most B such that every initial cluster of $\boldsymbol{A}$ with respect to C is of type (i), (iii), or (iv). Moreover, there is at most one initial cluster of type (iv).*

Proof. Because $(\boldsymbol{A}, k, B, \ell)$ is a yes-instance, there is a set of outliers O and a k-clustering of the remaining vectors of cost at most B. Among all such clusterings, we select a clustering $C = \{I_1, I_2, \ldots, I_k\}$ of $\{1, 2, \ldots, n\} \setminus O$ of cost at most B such that with respect to C,

$$\text{the number of initial clusters of type (ii) is minimum;} \tag{1}$$

subject to (1)

$$\text{the number of initial clusters of type (v) is minimum;} \tag{2}$$

and subject to (1) and (2),

$$\text{the number of initial clusters of type (iv) is minimum.} \tag{3}$$

We claim that C is the required clustering. Targeting toward a contradiction, assume first that there is an initial cluster of $\mathcal{J}$ of type (ii) with respect to C. Without loss of generality, let this cluster be J_1. Then J_1 is not fully contained in any cluster of C, but it is fully contained in the union of all clusters. We convert J_1 into a type (i) initial cluster by the following modification of C. Let $\boldsymbol{c}_i$ be the center of cluster I_i for $i \in \{1, \ldots, k\}$. Also, let $\mathbf{a}$ be the unique column corresponding to J_1 (the whole initial cluster consists of columns equal to $\mathbf{a}$). Let $\boldsymbol{c}_q$ be a center that minimizes the cost $d_H(\mathbf{a}, \boldsymbol{c}_i)$ over all centers $\boldsymbol{c}_i$. We construct a new clustering C' by reassigning the columns of J_1 to the cluster I_q. Thus, in the new clustering, J_1 becomes of type (i). The assignment of all other initial clusters remains the same. Since $d_H(\mathbf{a}, \boldsymbol{c}_q) \leq d_H(\mathbf{a}, \boldsymbol{c}_1)$, we have that the cost of the new

clustering C' is at most B, thus contradicting assumption (1). Hence, there are no initial clusters of $\mathcal{J}$ of type (ii) with respect to C.

We apply the same arguments to the non-outlier part of a type (v) initial cluster with respect to C. The only difference is that now the type (v) initial cluster becomes type (iv), thus contradicting assumption (2).

Finally, we already know that there are no type (ii) and (v) initial clusters with respect to C. Assume that there are at least two, say J_1 and J_2, initial clusters of type (iv) with respect to C. Then, there are clusters C_{i_1} and C_{i_2} in C such that $J_1 \subseteq C_{i_1} \cup O$ and $J_2 \subseteq C_{i_2} \cup O$. (We do not exclude the possibility of $i_1 = i_2$ here.) Let $\mathbf{c}_{i_1}$ and $\mathbf{c}_{i_2}$ be the centers of C_{i_1} and C_{i_2}, respectively. Let also $\mathbf{a}_1$ and $\mathbf{a}_2$ be the unique columns corresponding to J_1 and J_2. Without loss of generality, we assume that $d_H(\mathbf{a}_1, \mathbf{c}_{i_1}) \leq d_H(\mathbf{a}_2, \mathbf{c}_{i_2})$. There are two cases: (1) $|J_2 \cap C_{i_2}| \leq |J_1 \cap O|$ and (2) $|J_2 \cap C_{i_2}| > |J_1 \cap O|$. In the first case, we can assign additional $|J_2 \cap C_{i_2}|$ columns of J_i to C_{i_1} and move $|J_t \cap C_{i_2}|$ columns of J_2 from C_{i_2} to O. The cost increase is $|J_2 \cap C_{i_2}| \cdot (d_H(\mathbf{a}_1, \mathbf{c}_{i_1}) - d_H(\mathbf{a}_2, \mathbf{c}_{i_2})) \leq 0$. Also, the type of J_2 is now changed to (iii) with respect to the new clustering. Thus, the number of type (iv) initial clusters has decreased by at least 1 and no type (ii) or (iv) clusters were created. For the second case, assign the $|J_1 \cap O|$ columns of J_1 from O to C_{i_1} and move $|J_1 \cap O|$ more columns of J_2 from C_{i_2} to O. The cost increase is $|J_1 \cap O| \cdot (d_H(\mathbf{a}_1, \mathbf{c}_{i_1}) - d_H(\mathbf{a}_2, \mathbf{c}_{i_2})) \leq 0$. Also, the type of J_1 is now changed to (i) with respect to the new clustering. Thus, in this case also, the number of type (iv) initial clusters has decreased by 1. In both cases, we constructed clusterings contradicting (3). Hence, there is at most one initial cluster of type (iv) with respect to C. □

Now, we describe our algorithm. The algorithm tries to find a feasible clustering (if any) that satisfies the property of Lemma 5.1. As there is at most one initial cluster that can get split, we can guess it in advance. Suppose J_t is this initial cluster. We can also guess the number of columns ℓ' of J_t that would be part of the outliers set. We construct a new instance of the problem that contains all the columns in J_i for $1 \leq i \leq \beta$ and $i \neq t$, and exactly $|J_t| - \ell'$ columns of J_t. Let $\boldsymbol{A}'$ be the matrix $\boldsymbol{A}$ containing these columns. Then, our new instance is $(\boldsymbol{A}', k, B, \ell - \ell')$. The advantage of working with this instance is that in the desired clustering no initial cluster gets split. Next, we show how to find such a clustering for this instance.

For simplicity of notations, let us denote the input instance by $(\boldsymbol{A}, k, B, \ell)$. As no initial cluster in the desired clustering gets split, an initial cluster either can form its own cluster, becomes part of the outliers, or gets merged with other initial clusters to form a new cluster. We refer to the last type of clusters as *composite* clusters. Now, we have the following simple observation.

Observation 5.2. *In a feasible clustering, the total number of initial clusters that can be part of the composite clusters is at most $2B$. The total number of composite clusters is at most B.*

Next, we use the color coding technique of [2] to randomly color the initial clusters by one of the colors in $\{1, 2, \ldots, 2B\}$. The idea is that with sufficient probability the initial clusters that form the composite clusters get colored by pairwise distinct colors. Thus, assuming this event occurs, it is sufficient to select only one initial cluster from each color class to find out such initial clusters. However, we do not know how to combine those initial clusters. Nevertheless, we can guess such a combination by trying all possible partitions of $\{1, 2, \ldots, 2B\}$ that contain at most B parts. For each part, we can also guess the cost of the corresponding composite cluster. By Observation 5.2, the selected initial clusters corresponding to each part form a composite cluster. Each initial cluster that is not selected would either form its own cluster or become part of the outliers. However, an initial cluster that forms its own cluster does not incur any cost. Hence, any subsets of the appropriate number of non-selected initial clusters can form their own clusters without incurring any extra cost. However, we need to ensure that the number of outliers is at most ℓ. To ensure this,

we assign the non-selected initial clusters to the outliers set O in the non-decreasing order of their cardinalities. This ensures that we can pack the maximum number of initial clusters into O.

In light of the above discussion, the problem boils down to the following Restricted Clustering problem with exactly one cluster. Let S_j be the set of indices of the initial clusters colored by the color j for all $1 \leq j \leq 2B$. Also, denote the unique column in J_j by $\boldsymbol{b}^j$ for $1 \leq j \leq \beta$, and let the weight of $\boldsymbol{b}^j$, $w_j = |J_j|$. Fix a partition $T_1, T_2, \ldots, T_\tau$ of $\{1, 2, \ldots, 2B\}$. Also, guess τ positive integers $B_1, B_2, \ldots, B_\tau$ such that $\sum_{i=1}^{\tau} B_i = B$. B_i is the guess for the cost of the i^{th} composite cluster.

Fix a part i of the partition, where $1 \leq i \leq \tau$. Let the set of color indices $T_i = \{i_1, i_2, \ldots, i_p\}$. For each $1 \leq t \leq p$, let $U_t^i = \cup_{j \in S_{i_t}} \boldsymbol{b}^j$, i.e., the set of unique columns of the initial clusters that are colored by color i_t. The Restricted Clustering problem that we need to solve is the following. We are given the sets of columns $U_1^i, U_2^i, \ldots, U_p^i$ and a parameter B_i. The goal is to select p columns $\boldsymbol{b}^{j_1}, \boldsymbol{b}^{j_2}, \ldots, \boldsymbol{b}^{j_P}$ and a cluster center $\boldsymbol{s}^i$ such that $\boldsymbol{b}^{j_t} \in U_t^i$ for $1 \leq t \leq p$ and $\sum_{t=1}^{p} w_{j_t} \cdot d_H(\boldsymbol{b}^{j_t}, \boldsymbol{s}^i) \leq B_i$.

Next, we will show that Restricted Clustering can be solved in $2^{O(B_i \log B_i)} |\Sigma|^{B_i} (nm)^{O(1)}$ time using the hypergraph-based technique of Marx [28].

5.1 Solving Restricted Clustering

For simplicity, we drop the suffix i from all the notations. Thus, now we are given the sets of columns $U_1, U_2, \ldots, U_p$ and a parameter B. Let $\boldsymbol{c}$ be a center corresponding to an optimum feasible clustering $\mathcal{I}$ with the set $X = \{\boldsymbol{b}^{j_1}, \boldsymbol{b}^{j_2}, \ldots, \boldsymbol{b}^{j_P}\}$ of chosen columns. Also, let $U = \cup_{t=1}^{p} U_t$.

Suppose $\boldsymbol{x}$ is a vector such that there are at most B positions h with $\boldsymbol{x}[h] \neq \boldsymbol{c}[h]$. Consider the hypergraph H defined in the following way with respect to $\boldsymbol{x}$. The labels of the vertices of H are in $\{1, 2, \ldots, m\}$. For each $\boldsymbol{b}^{j_t} \in X$, we add w_{j_t} copies of the edge $S \subseteq \{1, 2, \ldots, m\}$ such that $h \in S$ if $\boldsymbol{x}[h] \neq \boldsymbol{b}^{j_t}[h]$.

Lemma 5.3. *Suppose the input is a yes-instance. Consider a vector $\boldsymbol{x}$ such that there is at most B positions h where $\boldsymbol{x}[h] \neq \boldsymbol{c}[h]$. Also, consider the hypergraph H defined above with respect to $\boldsymbol{x}$. There exists a subhypergraph $H_0^*(V_0^*, E_0^*)$ of H with the following properties:*

(1) $|V_0^*| \leq B$.
(2) $|E_0^*| \leq 200 \ln B$.
(3) *The indices in V_0^* are the exact positions h such that $\boldsymbol{x}[h] \neq \boldsymbol{c}[h]$.*
(4) *The fractional cover number of H_0^* 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 $\boldsymbol{x}[h] \neq \boldsymbol{c}[h]$. Let $H_0(V_0, E_0)$ be the subhypergraph of H induced by P. We show that H_0 satisfies the first, third, and fourth properties mentioned in Lemma 5.3.

By definition of $\boldsymbol{x}$, P contains at most B indices. The first property follows immediately. Next, we show that the third property also follows for H_0. As $V_0 \subseteq P$, in every position $h \in V_0$, $\boldsymbol{x}[h] \neq \boldsymbol{c}[h]$. It is sufficient to show that for any position h with $\boldsymbol{x}[h] \neq \boldsymbol{c}[h]$, $h \in V_0$. Consider any such position h. Then, if there is at least one $y \in X$ such that $\boldsymbol{x}[h] \neq \boldsymbol{y}[h]$, there is an edge e in H that contains h. Now, $h \in P$ by definition. Thus, the edge $e' = e \cap P \in H_0$ also contains h. Hence, $h \in V_0$. In the other case, for all $y \in X$, $\boldsymbol{x}[h] = \boldsymbol{y}[h]$. However, this case does not occur. Otherwise, we can replace the symbol $\boldsymbol{c}[h]$ by $\boldsymbol{x}[h]$. The new center $\boldsymbol{c}$ would incur lower cost, as now $\boldsymbol{c}[h] = \boldsymbol{y}[h]$ for all $\boldsymbol{y} \in X$ and previously $\boldsymbol{c}[h] \neq \boldsymbol{y}[h]$ for all $\boldsymbol{y} \in X$. Next, we show that the fourth property holds for H_0.

Lemma 5.4. *The fractional cover number of H_0 is at most* 2.

Proof. Note that the total number of edges of H_0 is $\tau = \sum_{t=1}^{p} w_{j_t}$. We claim that each vertex of H_0 is contained in at least $\tau/2$ edges.

Consider any vertex h of H_0. Let h be contained in $\delta\tau$ edges. Thus, by definition, $(1-\delta)\tau$ vectors of X (with multiplicity equal to their weights) do not differ from $\boldsymbol{x}$ at location h. Consider replacing $\boldsymbol{c}[h]$ by $\boldsymbol{x}[h]$ at position h of $\boldsymbol{c}$. Next, we analyze the change in the cost of the clustering $\mathcal{I}$. Note that the cost corresponding to positions other than h remains same. As $\boldsymbol{c}[h] \neq \boldsymbol{x}[h]$, previously each of the $(1-\delta)\tau$ columns were paying a cost of at least 1 corresponding to position h. Now, as $\boldsymbol{c}[h]$ is replaced by $\boldsymbol{x}[h]$ and consequently $\boldsymbol{c}[h] = \boldsymbol{x}[h]$, for these columns, the cost decrease is at least $(1-\delta)\tau$. Now, the cost increase for the remaining $\delta\tau$ columns is at most $\delta\tau$.

As $\mathcal{I}$ is an optimum feasible clustering, the cost decrease must be at most the cost increase. It follows that $\delta \geq 1/2$. □

So far we have proved the existence of a subhypergraph that satisfies all the properties except the second. Next, we prove the existence of a subhypergraph that satisfies all the properties. The proof of the following lemma is very similar to the proof of Lemma 4.3.

Lemma 5.5. *Let $B \geq 2$. Consider the subhypergraph H_0 that satisfies all the properties of Lemma 5.3 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 5.3.*

5.1.1 The Algorithm. We consider all possible columns $\boldsymbol{x} \in U$ and for each of them construct a hypergraph G whose vertices are in $\{1, 2, \ldots, m\}$. For each column $\boldsymbol{b}_j \in U$, we add w_j copies of the edge $S \subseteq \{1, 2, \ldots, m\}$ such that $h \in S$ if $\boldsymbol{x}[h] \neq \boldsymbol{b}_j[h]$. 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 3.1 to find every place P' where H_0 appears in G as a subhypergraph. For each such set P', we perform all possible $B' \leq B$ edits in $\boldsymbol{x}$ at the locations in P'. After each such edit, we retrieve the edited vector $\boldsymbol{x}$ and construct a clustering considering $\boldsymbol{x}$ as its center in the following way. For each $1 \leq t \leq p$, we select a vector $\boldsymbol{b}_j$ from U_t such that $d_H(\boldsymbol{x}, \boldsymbol{b}_j)$ is the minimum over all $\boldsymbol{b}_j \in U_t$. If the cost of this clustering is at most B, we output $\boldsymbol{x}$ as a candidate center. If no such vector is output as a candidate center, we return "NO."

Next, we analyze the correctness and time complexity of this algorithm.

5.1.2 The Analysis.

Lemma 5.6. *The above algorithm successfully outputs $\boldsymbol{c}$ as a candidate center.*

Proof. Consider any vector $\boldsymbol{x} \in B$. Note that there are at most B positions h where $\boldsymbol{x}[h] \neq \boldsymbol{c}[h]$. Consider the hypergraph G constructed by the algorithm corresponding to $\boldsymbol{x}$. Note that the hypergraph H defined in Lemma 5.3 is a partial subgraph of G. Thus, the subhypergraph H_0^* 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_0^* must be considered by the algorithm. Let P' be the place in G where H_0^* appears. By the third property of Lemma 5.3, the locations in P' are the exact positions h such that $\boldsymbol{x}[h] \neq \boldsymbol{c}[h]$. It follows that an edit corresponding to P' generates the vector $\boldsymbol{c}$, as $d_H(\boldsymbol{x}, \boldsymbol{c}) \leq B$. Hence, $\boldsymbol{c}$ must be an output of the algorithm. □

5.1.3 Time Complexity. Next, we discuss the time complexity of our algorithm. For each choice of $\boldsymbol{x}$, the hypergraph G can be constructed in $n^{O(1)}$ time. The number of distinct hypergraphs H_0 with at most B vertices and at most $200 \ln B$ edges is $2^{O(B \log B)}$ 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 vector $\boldsymbol{y} \in B$, $d_H(\boldsymbol{y}, \boldsymbol{c}) \leq B$. By triangle inequality, $d_H(\boldsymbol{x}, \boldsymbol{y}) \leq 2B$. 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 3.1, it follows that every occurrence of H_0 in G can be found in $B^{O(B)} \cdot (2B)^{4B+1} \cdot n^5 \cdot m^2 = B^{O(B)} \cdot n^{O(1)} m^2$ time. If H_0 appears at some place in G, it would take $O((B|\Sigma|)^B)$ time to edit $\boldsymbol{x}$. Hence, in total the algorithm runs in $2^{O(B \log B)}|\Sigma|^B \cdot (nm)^{O(1)}$ time.

Finally, Theorem 2.8 is proven. We restate it here for convenience.

Theorem 2.8. *k-Clustering with Column Outliers is solvable in time $2^{O(B \log B)}|\Sigma|^B \cdot (nm)^{O(1)}$.*

6 CLUSTERING WITH ROW OUTLIERS

Here we show that Feature Selection is a special case of Constrained Clustering with Outliers.

Lemma 2.5. *For any instance $(\boldsymbol{D}, k, B, \ell)$ of Feature Selection, one can construct in time $O(mn + k \cdot |\Sigma|^k)$ 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 $\ell' = \ell$.*

Proof. Given an instance $I = (\boldsymbol{D}, k, B, \ell)$ of Feature Selection, we construct an instance $I' = (\boldsymbol{A}, k', B', \ell', \mathcal{R})$ of Constrained Clustering with Outliers such that $\boldsymbol{A}$ is the transpose of $\boldsymbol{D}$, $k' = |\Sigma|^k$, $B' = B$, and $\ell' = \ell$. The set of constraints $\mathcal{R}$ is defined as follows. Let $S = \{s_1, s_2, \ldots, s_{|\Sigma|^k}\}$ be the lexicographic ordered collection of all $|\Sigma|^k$ strings of length k on alphabet Σ. Also, let Q be the $|\Sigma|^k \times k$ matrix such that the ith row of Q is the ith string of S. Thus, each element of Q is in Σ. For each column $\boldsymbol{q}^j$ of Q with $1 \leq j \leq k$, we construct a tuple $z_j = (\boldsymbol{q}^j[1], \boldsymbol{q}^j[2], \ldots, \boldsymbol{q}^j[|\Sigma|^k])$. Let R be the set of these tuples. To construct $\mathcal{R}$ we set $R_i = R$ for all i. This finishes our construction.

Next, we prove that I is a yes-instance of Feature Selection if and only if I' is a yes-instance of Constrained Clustering with Outliers. The lemma follows immediately.

First, suppose I' is a yes-instance of Constrained Clustering with Outliers. Let $X = \{X_1, X_2, \ldots, X_{|\Sigma|^k}\}$ be a feasible clustering of the columns of $\boldsymbol{A} \setminus O$, the matrix $\boldsymbol{A}$ after removing the columns in O, where O is the set of outliers. Also, let $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{|\Sigma|^k}$ be the corresponding centers. Note that as this clustering satisfies the constraints in $\mathcal{R}$, each tuple $(\boldsymbol{c}_1[j], \boldsymbol{c}_2[j], \ldots, \boldsymbol{c}_{|\Sigma|^k}[j])$ is one of the k tuples of R. Let $Y = \{Y_1, Y_2, \ldots, Y_k\}$ be the partition of the rows of $\boldsymbol{A} \setminus O$, such that Y_t contains all the indices j such that $(\boldsymbol{c}_1[j], \boldsymbol{c}_2[j], \ldots, \boldsymbol{c}_{|\Sigma|^k}[j]) = z_t$ for $1 \leq t \leq k$. We reorder the rows of matrix $\boldsymbol{A} \setminus O$ based on this partition. In particular, we modify the row numbers in a way so that the rows in Y_1 become the first $|Y_1|$ rows, the rows in Y_2 become the next $|Y_2|$ rows, and so on (see Figure 1). Now, let us go back to the clustering $\{X_1, X_2, \ldots, X_{|\Sigma|^k}\}$. Note that this is still a feasible clustering of the new matrix $\boldsymbol{A} \setminus O$ with cost at most B with rows of the centers reordered in the same way. Indeed, by construction, the reordering of the centers ensures that the center of X_i is the column vector $(z_1[i], \ldots |Y_1| \text{ times }, z_2[i], \ldots |Y_2| \text{ times }, \ldots, z_k[i], \ldots |Y_k| \text{ times})$ (see Figure 1). Now, note that $\{Y_1, Y_2, \ldots, Y_k\}$ is a clustering of the columns in $\boldsymbol{D}^{-O}$. We claim that its cost is at most B. As $|O| \leq \ell$, it follows that I is a yes-instance of Feature Selection. Note that some clusters might be empty. But, as they do not incur any cost, we keep them for simplicity. To analyze the cost, set the center of Y_j to the column vector $(z_j[1], \ldots |X_1| \text{ times }, z_j[2], \ldots |X_2| \text{ times }, \ldots, z_j[|\Sigma|^k], \ldots |X_{|\Sigma|^k}| \text{ times})$ (see Figure 1). Now, because of the symmetry between the two clusterings X and Y, their costs are the same. Hence, the claim follows.

Now, suppose I is a yes-instance of Feature Selection and consider a feasible clustering $Y = \{Y_1, Y_2, \ldots, Y_k\}$. Let O be the set of indices of the outliers. Also, let $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k$ be the corresponding centers. Note that each string of the form $\boldsymbol{c}_1[j]\boldsymbol{c}_2[j] \ldots \boldsymbol{c}_k[j] \in S$. For a string $s \in S$, we say row j is of type s if $\boldsymbol{c}_1[j]\boldsymbol{c}_2[j] \ldots \boldsymbol{c}_k[j] = s$. Let $\{X_1, X_2, \ldots, X_{|\Sigma|^k}\}$ be the partition of the rows of $\boldsymbol{D}^{-O}$ based on their types. In particular, for the ith string $s \in S$, X_i contains all rows of type s. Note that $\{X_1, X_2, \ldots, X_{|\Sigma|^k}\}$ is also a clustering of the columns of $\boldsymbol{A} \setminus O$. We show that the cost

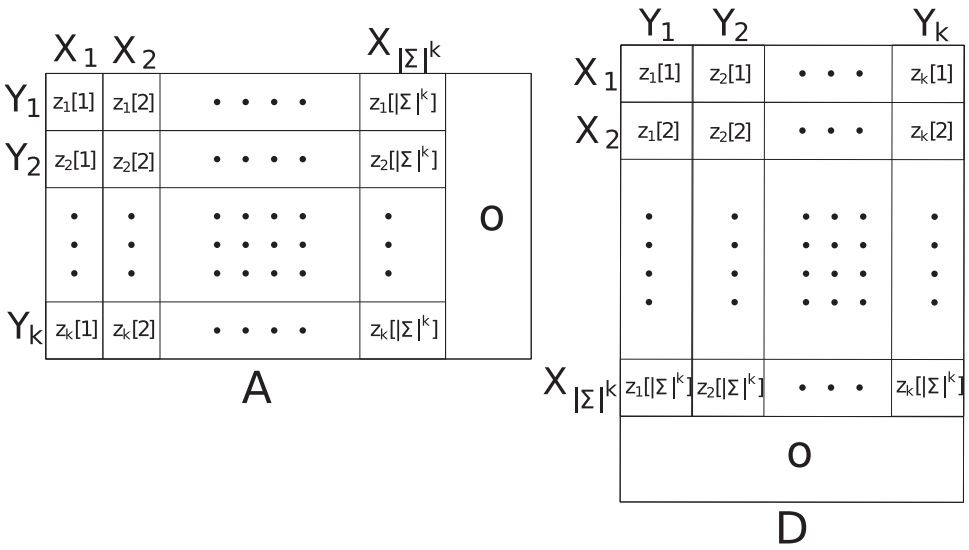


Fig. 1. Figure showing the clustering X of $\boldsymbol{A} \setminus O$ and Y of $\boldsymbol{D}^{-O}$. The common center coordinate values are shown inside submatrices.

of this clustering is at most B. To analyze the cost, choose the column vector $(s_i[1], \ldots |Y_1|$ times $, s_i[2], \ldots |Y_2|$ times $, \ldots, s_i[k], \ldots |Y_k|$ times) as the center vector of the cluster X_i of the columns of $\boldsymbol{A} \setminus O$. Note that the centers chosen in this way satisfy the constraints in $\mathcal{R}$, as the center tuple $(s_1[j], s_2[j], \ldots, s_{|\Sigma|^k}[j]) = (\boldsymbol{q}^j[1], \boldsymbol{q}^j[2], \ldots, \boldsymbol{q}^j[|\Sigma|^k])$ corresponding to index j is in R. It follows that the cost of this clustering is at most B, as it is induced by the clustering Y and the centers $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k$. □

Lemma 2.5 together with Theorem 2.4 immediately give us Theorem 2.1 with $f(k, B, |\Sigma|) = (k'B)^{O(k'B)}|\Sigma|^{k'B}$ and $g(k, |\Sigma|) = O(k')$, where $k' = |\Sigma|^k$. The theorem is restated here.

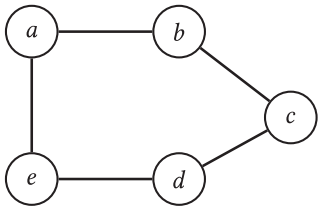
Theorem 2.1. *Feature Selection is solvable in time $f(k, B, |\Sigma|) \cdot m^{g(k,|\Sigma|)} \cdot n^2$, where f and g are computable functions.*

7 LOWER BOUNDS

In this section, we present the lower-bound results. First, we recall a few complexity notions essential to our hardness proofs.

A *parameterized problem* is a language $Q \subseteq \Sigma^* \times \mathbb{N}$, where Σ^* is the set of strings over a finite alphabet Σ. Respectively, an input of Q is a pair (I, k), where $I \subseteq \Sigma^*$ and $k \in \mathbb{N}$; k is the *parameter* of the problem. A parameterized problem Q is *fixed-parameter tractable* (FPT) if it can be decided whether $(I, k) \in Q$ in time $f(k) \cdot |I|^{O(1)}$ for some function f that depends on the parameter k only. Respectively, the parameterized complexity class FPT is composed by fixed-parameter tractable problems. The W-hierarchy is a collection of computational complexity classes: we omit the technical definitions here. The following relation is known among the classes in the W-hierarchy: FPT = W[0] ⊆ W[1] ⊆ W[2] ⊆ ... ⊆ W[P]. It is widely believed that FPT ≠ W[1], and hence if a problem is hard for the class W[i] (for any $i \geq 1$), then it is considered to be fixed-parameter intractable. We refer to books [13, 15] for a detailed introduction to parameterized complexity.

(a) The input graph G, $t = 2$.



(b) The corresponding 5×5 matrix of the clustering problem. The number of rows to remove is 3, the number of clusters is 3.

	ab	bc	cd	de	ae
a	1	0	0	0	1
b	1	1	0	0	0
c	0	1	1	0	0
d	0	0	1	1	0
e	0	0	0	1	1

(c) A selection of inliers achieving a zero-cost clustering.

b	1	1	0	0	0
d	0	0	1	1	0

(d) A selection of inliers with a nonzero-cost of clustering.

d	0	0	1	1	0
e	0	0	0	1	1

Fig. 2. An illustration of the reduction in Theorem 2.2 showing a possible input graph, the obtained matrix, and two possible selections of the inliers.

We also make use of the following complexity hypothesis formulated by Impagliazzo et al. [23]:

> **Exponential Time Hypothesis (ETH)**: There is a positive real s such that 3-CNF-SAT with n variables and m clauses cannot be solved in time $2^{sn} \cdot (n+m)^{O(1)}$.

Our lower bounds hold for the Feature Selection problem, and even for its special case where $B = 0$. This case may be viewed as the problem of partitioning columns of the input matrix into clusters where columns in each cluster are identical, except for the outlier rows. We show that this exact version of Feature Selection is W[1]-hard when parameterized by the number of clusters k and the number of inliers $(n - \ell)$, and also W[1]-hard when parameterized by the number of outliers ℓ.

The first lower bound shows that the result in Theorem 2.1 is tight in the sense that k must be a constant. This is in contrast to the k-Clustering with Column Outliers problem, for which we show the FPT algorithm for the stronger parameterization (Theorem 2.8). Thus, Feature Selection is fundamentally more difficult than k-Clustering with Column Outliers. Also, note that k-Clustering with Column Outliers is trivially solvable in polynomial time when the cost B is zero.

Lemma 7.1. *Feature Selection is W[1]-hard when parameterized by $k + (m - \ell)$ when $B = 0$ and $\Sigma = \{0, 1\}$. Assuming ETH, there is no algorithm solving the problem with $B = 0$ and the binary alphabet in time $m^{o(k)} \cdot n^{O(1)}$.*

Proof. We show a reduction from the Independent Set problem. Given a graph $G(V, E)$ and a parameter t, the task in Independent Set is to determine whether there is an independent set of size at least t in G. Consider an instance $(G(V, E), t)$ of Independent Set. Assume that for any t-sized subset S of V, each of s in S is adjacent to a vertex in $V \setminus S$, and there is an edge inside $V \setminus S$. Any graph G can be tweaked to meet this assumption by adding a $(t + 2)$-sized clique C to the graph and connecting each vertex of G to each vertex of C. Note that no vertex of C can be in any independent set of size at least two, and any independent set of the original graph is present in the newly constructed graph. Thus, the new instance of Independent Set is equivalent to the original one, and it satisfies the assumption stated.

From (G, t), we construct an instance of Feature Selection over the alphabet $\Sigma = \{0, 1\}$. The input matrix $\boldsymbol{A}$ is the incidence matrix of G where rows are indexed by vertices and columns by edges. Set $k = t + 1$, $\ell = |V| - t$, $B = 0$. See Figure 2 for an example.

Now it remains to show the correctness of the reduction. Assume there is an independent set I of size t in G. Let the inliers be the rows corresponding to I, and O be the set of the remaining outlier rows. Restricted to the inlier rows, every column of $\boldsymbol{A}$ either is a zero column or has a single

one since no edge of G has both ends in I. Thus, there are at most $t+1$ distinct columns in $\boldsymbol{A}^{-O}$, so there is a zero-cost $(t+1)$-clustering. The centroids in this clustering are the following t-columns: the zero column and all the t possible columns with a single one.

For the other direction of the correctness proof, assume there is a solution to the constructed Feature Selection instance. Consider the set I of the vertices corresponding to the inlier rows, and the set O of the outlier rows. By the assumption on G, every vertex in I is adjacent to a vertex in $V \setminus I$, and there is an edge inside $V \setminus I$. Thus, $\boldsymbol{A}^{-O}$ has a zero column and all t distinct columns have a single one. If there is an edge e inside I, then there are at least $t+2$ distinct columns in the matrix since even after removing the rows of O, the column corresponding to e has two ones. In this case, no zero-cost clustering with $t+1$ clusters is possible. Thus, I must be an independent set.

Thus, the reduction is valid. It is well known that Independent Set is W[1]-hard, and also not solvable in time $f(t) \cdot |V|^{o(t)}$ for any computable function f unless ETH fails. Observing that $k = t+1$ and $m - \ell = t$ concludes the proof of the lemma. □

Since Lemma 2.5 gives a parameterized reduction from Feature Selection to Constrained Clustering with Outliers, we immediately get the following corollary.

Corollary 7.2. *Constrained Clustering with Outliers is W[1]-hard when parameterized by* $k + (n - l)$ *when* $B = 0$ *and* $\Sigma = \{0, 1\}$.

We do not get the analogous ETH bound, however, as the number of clusters in the constructed instance of Constrained Clustering with Outliers is exponential.

The second lower bound shows that bounding both the number of outliers and the cost B is insufficient for an FPT algorithm as well.

Lemma 7.3. *Feature Selection is W[1]-hard when parameterized by* ℓ *when* $B = 0$ *and* $\Sigma = \{0, 1\}$. *Assuming ETH, there is no algorithm solving the problem with* $B = 0$ *and the binary alphabet in time* $m^{o(\ell)} \cdot n^{O(1)}$.

Proof. We show a reduction from the Partial Vertex Cover problem. The input to Partial Vertex Cover is a graph $G(V, E)$ and numbers t, q. The problem is to decide whether there is a subset $C \subset V$ of size at most t such that at least q edges of G are covered by C. The Partial Vertex Cover problem is well known to be as hard as Independent Set ([13], Theorem 13.6). That is, when Partial Vertex Cover is parameterized by t, there is a parameter-preserving reduction from Independent Set.

Consider an instance (G, t, q) of Partial Vertex Cover. The idea of the reduction is similar to that in Lemma 7.1. In short, we argue that choosing t vertices in G in the way that maximizes the number of covered edges corresponds to removing t rows in the incidence matrix of G in the way that minimizes the number of distinct columns, up to a certain assumption on G. Next, we show that formally.

First, we modify G to obtain a new graph $G'(V', E')$. Let P be a set of two new vertices, and D be a set of d new vertices, where $d = 5 + t + |E| - q$. Set V' to $V \cup P \cup D$. Keep all edges of G on V, and connect each vertex of P to all other vertices of G', including the other vertex of P. That is,

$$E' = E \cup \{pv \mid p \in P, v \in V' \setminus \{p\}\} .$$

The total number of edges in G' is $|E| + 2 \cdot |V'| - 3$.

The graph G' behaves in the same way as G with respect to Partial Vertex Cover, as stated in the following claim. Set $t' = t + 2$ and $q' = q + 2 \cdot |V'| - 3$.

Claim 7.4. *The instance* (G, t, q) *is a yes-instance of Partial Vertex Cover if and only if* (G', t', q') *is a yes-instance of Partial Vertex Cover.*

Proof. First, assume there is a subset $C \subset V$ of size at most t covering at least q edges of G. Set $C' = C \cup P$, $|C'| \leq t + 2 = t'$, and vertices of P cover the additional $2 \cdot |V'| - 3$ edges that are not present in G.

For the other direction, assume there is a subset $C' \subset V'$ of size at most t' covering at least q' edges of G'. We may assume that $P \subset C'$; otherwise we may replace any vertex of C by a vertex of P, and the number of covered edges does not decrease since vertices in P are adjacent to all vertices. Now, $C = C' \cap V$ is a solution for (G, t, q): $|C| \leq |C' \setminus P| \leq t$, and since $|E' \setminus E| = q' - q$, C must cover at least q edges in G. □

Now we construct the input instance $(\boldsymbol{A}, k, B, \ell)$ of Feature Selection over the alphabet $\Sigma = \{0, 1\}$. The input matrix $\boldsymbol{A}$ is the incidence matrix of G' where rows are indexed by vertices and columns by edges. Set $k = 1 + |V'| - t' + |E'| - q'$, $\ell = t'$, $B = 0$.

To show the correctness of the reduction, first assume there is a set $C \subset V'$ of size t' covering at least q' edges in G'. Let O be the set of rows in $\boldsymbol{A}$ corresponding to C. We claim that there are at most $1 + (|V'| - t') + (|E'| - q')$ distinct columns in $\boldsymbol{A}^{-O}$, that is, at most one zero column, at most $|V'| - t'$ columns with a single one corresponding to the vertices of $V' \setminus C$ adjacent to C, and at most $|E'| - q'$ columns with two ones corresponding to the edges not covered by C. Since there are at most k distinct columns, the cost of k-clustering is zero if we remove the rows of O.

In the other direction, assume there is a solution to the constructed Feature Selection instance, that is, a set O of at most t' rows such that $\boldsymbol{A}^{-O}$ has at most k distinct columns. Consider the set C of vertices corresponding to the outlier rows O. First, we show that C must contain P. If this does not hold, either $C \cap P$ is empty or $|C \cap P| = 1$. Assume $C \cap P$ is empty; in this case out of $2 \cdot |V'| - 3$ edges incident to P at most $2t'$ are covered by vertices of C. The uncovered of these edges correspond to at least $2 \cdot |V'| - 3 - 2t'$ columns of $\boldsymbol{A}$ that have still two ones each in $\boldsymbol{A}^{-O}$, and thus they are all distinct. However, the number of clusters is smaller than the number of such columns since

$$2 \cdot |V'| - 3 - 2t' = |V'| - 2t' - 1 + |V| + d > 1 + |V'| - t' + |E'| - q' = k,$$

as $d = 3 + t' + |E| - q$ and $|E| - q = |E'| - q'$. This is a contradiction to O being a solution.

In the other case, $|C \cap P| = 1$, denote by p the vertex of P that is in C, and by p' the other vertex of P. At most t' vertices in $V' \setminus \{p\}$ are in C, and thus there are at least $|V'| - 1 - t'$ distinct columns in $\boldsymbol{A}^{-O}$ that correspond to edges from p to $V' \setminus C$; they each have a single one in the row corresponding to the second endpoint. There are also at least $|V'| - 2 - t'$ distinct columns in $\boldsymbol{A}^{-O}$ that correspond to edges from p' to $V' \setminus C$; they each have two ones in the rows corresponding to p' and the second endpoint. Thus, there are at least $2 \cdot |V'| - 3 - 2t'$ distinct columns in $\boldsymbol{A}^{-O}$, and we have already shown that this quantity is strictly larger than k, leading to the contradiction.

Now that we have shown $P \subset C$, we argue that all possible columns with less than two ones are present in $\boldsymbol{A}^{-O}$, giving an exact bound on the number of distinct columns with two ones. Formally, since $P \subset C$, there is a zero column in $\boldsymbol{A}^{-O}$ corresponding to the edge inside P. Take any $p \in P$; since $p \in C$ and any vertex v in $V' \setminus C$ is adjacent to p, there is a column in $\boldsymbol{A}^{-O}$ corresponding to the edge vp that has a single one in the row corresponding to v. This gives $|V'| - t'$ distinct columns with single ones, one for each vertex of $V' \setminus C$. Thus, there must be at most $k - 1 - (|V'| - t') = |E'| - q'$ distinct columns with two ones in $\boldsymbol{A}^{-O}$. These columns correspond exactly to the uncovered by C edges of G'. So C covers at least q' edges, and (G', t', q') is a yes-instance. This finishes the correctness proof, and from the hardness of Partial Vertex Cover the lemma follows. □

Clearly, Theorem 2.2, restated next for convenience, follows from Lemma 7.1 together with Lemma 7.3.

Theorem 2.2. *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 (ETH), the problem cannot be solved in time* $f(k) \cdot m^{o(k)} \cdot n^{O(1)}$ *for any function* f, *even when* $B = 0$ *and the alphabet* Σ *is binary.*

8 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 $2^{O(B \log B)}|\Sigma|^B \cdot (nm)^{O(1)}$ 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 $m^{o(k)} \cdot n^{O(1)}$ 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 $(kB)^{O(kB)}|\Sigma|^{kB} \cdot m^2 n^{O(k)}$. 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 and Robust Low Boolean-Rank Approximation 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 $k = 2$ [14, 22] and W[1]-hard when parameterized by B [20]. 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 from the Even Set problem [6]. The most interesting open question, in 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. 2013. Feature selection for clustering: A review. In *Data Clustering: Algorithms and Applications*, Charu C. Aggarwal and Chandan K. Reddy (Eds.). CRC Press, 30–373.

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

[3] Dana Angluin and Leslie G. Valiant. 1977. Fast probabilistic algorithms for Hamiltonian circuits and matchings. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing*. 30–41.

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

[5] Aditya Bhaskara and Srivatsan Kumar. 2018. Low rank approximation in the presence of outliers. In *APPROX/RANDOM'18*, Vol. 116, 4:1–4:16. https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2018.4

[6] Arnab Bhattacharyya, Édouard Bonnet, László Egri, Suprovat Ghoshal, Karthik C. S., Bingkai Lin, Pasin Manurangsi, and Dániel Marx. 2021. Parameterized intractability of even set and shortest vector problem. *J. ACM* 68, 3 (2021), 16:1–16:40. https://doi.org/10.1145/3444942

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

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

[9] Thierry Bouwmans, Necdet Serhat Aybat, and El-hadi Zahzah. 2016. *Handbook of Robust Low-rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. Chapman and Hall/CRC.

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

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

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

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

[14] Chen Dan, Kristoffer Arnsfelt Hansen, He Jiang, Liwei Wang, and Yuchen Zhou. 2015. On low rank approximation of binary matrices. *CoRR* abs/1511.01699 (2015). arXiv:1511.01699. http://arxiv.org/abs/1511.01699

[15] Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity*. Springer. https://doi.org/10.1007/978-1-4471-5559-1

[16] Uriel Feige. 2014. NP-hardness of hypercube 2-segmentation. *CoRR* abs/1411.0821 (2014). arXiv:1411.0821. http://arxiv.org/abs/1411.0821

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

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

[19] Fedor V. Fomin, Petr A. Golovach, and Kirill Simonov. 2021. Parameterized k-clustering: Tractability Island. *J. Comput. Syst. Sci.* 117 (2021), 50–74. https://doi.org/10.1016/j.jcss.2020.10.005

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

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

[22] Nicolas Gillis and Stephen A. Vavasis. 2015. On the complexity of robust PCA and ℓ_1-norm low-rank matrix approximation. *CoRR* abs/1509.09236 (2015). http://arxiv.org/abs/1509.09236

[23] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. 2001. Which problems have strongly exponential complexity. *J. Comput. Syst. Sci.* 63, 4 (2001), 512–530.

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

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

[26] Tao Li. 2005. A general model for clustering binary data. In *KDD'05*. 188–197.

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

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

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

[30] Pauli Miettinen and Stefan Neumann. 2020. Recent developments in Boolean matrix factorization. In *IJCAI' 20*. ijcai.org, 4922–4928.

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

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

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

[34] René Vidal, Yi Ma, and S. Shankar Sastry. 2016. *Generalized Principal Component Analysis (Interdisciplinary Applied Mathematics)*, Vol. 40. Springer. https://doi.org/10.1007/978-0-387-87811-9

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

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

Received 20 August 2021; revised 06 June 2023; accepted 23 June 2023