Fixed-Parameter and Approximation Algorithms for PCA with Outliers

Yogesh Dahiya [*1] **Fedor Fomin** [*2] **Fahad Panolan** [*3] **Kirill Simonov** [*2]

Abstract

PCA WITH OUTLIERS is the fundamental problem of identifying an underlying low-dimensional subspace in a data set corrupted with outliers. A large body of work is devoted to the information-theoretic aspects of this problem. However, from the computational perspective, its complexity is still not well-understood. We study this problem from the perspective of parameterized complexity by investigating how parameters like the dimension of the data, the subspace dimension, the number of outliers and their structure, and approximation error, influence the computational complexity of the problem. Our algorithmic methods are based on techniques of randomized linear algebra and algebraic geometry.

1. Introduction

Problem statement. We study a fundamental problem in unsupervised machine learning called PCA WITH OUTLIERS: given a set of n points in $\mathbb{R}^d$ with (unknown) k outliers, the problem is to identify the outliers so that the remaining set of points fits best into an unknown r-dimensional subspace. Without outliers, this is the classical *principal component analysis* (PCA), one of the most popular approaches of reducing the dimensionality of the data (Pearson, 1901; Hotelling, 1933; Eckart & Young, 1936). PCA is often formulated as the problem of finding the best low-rank approximation for a data matrix $\mathbf{A}$ by solving

$$\begin{aligned} &\text{minimize } \|\mathbf{A}-\mathbf{L}\|_F^2 \\ &\text{subject to } \operatorname{rank}(\mathbf{L}) \le r. \end{aligned}$$

Here $||\mathbf{A}||_F^2 = \sum_{i,j} a_{ij}^2$ is the squared Frobenius norm of the matrix $\mathbf{A}$. By the Eckart-Young theorem, see (Eckart & Young, 1936), PCA is efficiently solvable via Singular Value Decomposition (SVD). PCA is used as a preprocessing step in a great variety of modern applications including face recognition, data classification, and analysis of social networks.

However, PCA is very sensitive to outliers, and the presence of even a small number of outliers can lead to misleading conclusions, see (Croux & Haesbroeck, 2000). Outliers detection and PCA with robustness to outliers are the fundamental topics in Machine Learning and Robust Statistics. Chapter 3.3 of (Vidal et al., 2016) is a nice introduction to basic approaches to this problem. The handbook of (Bouwmans et al., 2016) on Robust Low-Rank decomposition provides a thorough overview of different variants and applications of robust PCA. PCA with outliers is strongly related to another well-studied problem, namely ROBUST SUBSPACE RECOVERY. Here the task is to identify a subspace that contains a large portion of data points. See the survey of (Lerman & Maunu, 2018) for an overview of ROBUST SUBSPACE RECOVERY.

In this paper, we study the mathematical model for PCA with outliers proposed in (Wright et al., 2009; Xu et al., 2010; Chen et al., 2011; Simonov et al., 2019). We do not make any assumptions on the distribution of outliers, and the model covers the worst-case scenarios of adversarial outliers.

PCA WITH OUTLIERS

Input: Data matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $r, k \in \mathbb{N}$.
Task:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{A}-\mathbf{L}-\mathbf{N}\|_F^2 \\ \text{subject to} \quad & \mathbf{L}, \mathbf{N} \in \mathbb{R}^{n\times d}, \operatorname{rank}(\mathbf{L}) \le r, \\ & \text{and } \mathbf{N} \text{ has at most} \\ & k \text{ non-zero rows.} \end{aligned}$$

PCA WITH OUTLIERS has a natural geometric interpretation. Given n points in $\mathbb{R}^d$, represented by the rows of $\mathbf{A}$, we seek for a set of k outliers, represented by the non-zero rows of $\mathbf{N}$, whose removal from $\mathbf{A}$ leaves the remaining $n-k$ inliers as close as possible to an r-dimensional subspace. The matrix $\mathbf{L}$ contains then the orthogonal projections of the inliers onto this subspace.

[*] Equal contribution [1] The Institute of Mathematical Sciences (HBNI), Chennai, India [2] Department of Informatics, University of Bergen, Norway [3] Department of Computer Science and Engineering, IIT Hyderabad, Hyderabad, Telangana, India. Correspondence to: Yogesh Dahiya <yogeshdahiya@imsc.res.in>, Kirill Simonov <kirillsimonov@gmail.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

(Simonov et al., 2019) proved a lower bound which (assuming ETH[1]) rules out any constant-factor approximation of PCA WITH OUTLIERS in time $f(d) \cdot n^{o(d)}$, for any function f of d. They also gave an algorithm solving PCA WITH OUTLIERS in time roughly $n^{\mathcal{O}(r \cdot d)}$. This is the point of departure for our work. Running time $n^{\mathcal{O}(r \cdot d)}$ of the algorithm of Simonov et al. is polynomial only when the input dimension d is a constant. However, in many PCA applications, the input dimension d is very large, and it is desirable to have algorithms that run in polynomial time when the target dimension r, that is, the dimension of the best-fit subspace, is small. Taking into account the computational lower bound from (Simonov et al., 2019), the existence of a polynomial-time algorithm for PCA WITH OUTLIERS is highly unlikely. However, this lower bound does not rule out existence of algorithms that run in polynomial time for small values of r. These are exactly the algorithms we seek in this paper and our work is driven by the following question: Whether it is possible to solve PCA WITH OUTLIERS in polynomial time for small values of r?

We address this question by making use of parameterized complexity. We refer to the book of (Cygan et al., 2015) for an introduction. The main idea of parameterized complexity is to measure the success of the algorithm in terms of both input size as well as one or several parameters that capture the structural properties of the instance. According to Tim Roughgarden (Roughgarden, 2020), the worst-case analysis takes a ”Murphy’s Law” approach to algorithm analysis. However, it is *never* the case that the input size is the *only* aspect of the input instance that affects the running time of an algorithm. For this reason, the running time estimates from a worst-case analysis are overly pessimistic. Our work sheds some light on the influence of parameters like the dimension of the data, the subspace dimension, the number of outliers and their structure, and approximation error, on the computational complexity of the problem.

Our results. We provide several algorithms for PCA WITH OUTLIERS that work in polynomial time for small values of r. Our first algorithmic result shows that if we allow a small approximation, then the problem is solvable in polynomial time when r is a constant.

Theorem 1.1. *For every $\varepsilon > 0$, an $(1 + \varepsilon)$-approximate solution to* PCA WITH OUTLIERS *can be found in time $n^{\mathcal{O}(\frac{r \log r}{\varepsilon^2})} \cdot d^{\mathcal{O}(1)}$.*

In other words, PCA WITH OUTLIERS admits a randomized Polynomial Time Approximation Scheme (PTAS) when the dimension r of the solution subspace is a fixed constant. According to (Simonov et al., 2019), unless ETH fails, PCA WITH OUTLIERS does not admit a constant-factor approximation in time $f(d) \cdot n^{o(d)}$, for any function f of d. Since $r \leq d$, up to the $\log r$ factor in the exponent of n, the running time of our PTAS is tight.

We also provide algorithms for solving PCA WITH OUTLIERS exactly. While the nature of outliers can be elusive, we make two assumptions on outliers. Both assumptions try to capture the property that outlier points are further from the best-fit subspace than inlier points. It appears that these assumptions can be very useful from the algorithmic perspective. The intuition behind the first assumption is that every outlier is further from an optimal solution than any inlier. The second assumption is stronger, it assumes that the squared distance from any outlier to the solution subspace is larger than the sum of all inliers’ squared distances to the subspace.

Definition 1.1 (**α-gap and α-heavy assumptions**). *For $\alpha > 0$, the α*-gap assumption *about instance $(\mathbf{A}, r, k)$ of* PCA WITH OUTLIERS *is that there is an optimal solution $(\mathbf{L}, \mathbf{N})$ with the following properties. Let O be the indices of outliers, that is the indices of non-zero rows of $\mathbf{N}$, and let $I = [n] \setminus O$ be the indices of inliers. Then for every $i \in O$ and $j \in I$,*

$$\operatorname{dist}^2(\mathbf{a_{i:}}, \mathbf{V}^*) > (1 + \alpha) \cdot \operatorname{dist}^2(\mathbf{a_{j:}}, \mathbf{V}^*),$$

where $\mathbf{V}^*$ is the r-dimensional subspace which spans the rows of $\mathbf{L}$ and $\mathbf{a_{i:}}$ denotes the i-th row of $\mathbf{A}$. Similarly, the α-heavy assumption *about $(\mathbf{A}, r, k)$ is that for every $i \in O$,*

$$\operatorname{dist}^2(\mathbf{a_{i:}}, \mathbf{V}^*) > (1 + \alpha) \sum_{j \in I} \operatorname{dist}^2(\mathbf{a_{j:}}, \mathbf{V}^*)$$

Note that α-gap and α-heavy assumptions capture a very wide class of instances of PCA WITH OUTLIERS, as they do not require any particular structure on the outliers, only that the outliers are farther from the optimal subspace by at least some small factor compared to the inliers. In particular, ROBUST SUBSPACE RECOVERY, an extremely well-studied problem in Robust Statistics (see (Lerman & Maunu, 2018) for further references), satisfies these assumptions. Let us remind, that the input to ROBUST SUBSPACE RECOVERY is an instance $(\mathbf{A}, r, k)$ of PCA WITH OUTLIERS, and the task is to decide whether there exist matrices $\mathbf{L}$ and $\mathbf{N}$ such that $\mathbf{A} = \mathbf{L} + \mathbf{N}$, the rank of $\mathbf{L}$ is at most r, and $\mathbf{N}$ has at most k non-zero rows. In ROBUST SUBSPACE RECOVERY all inliers lie in $\mathbf{V}^*$ and thus satisfy both α-gap and α-heavy assumptions. Thus both “α-assumptions” create interesting parameterized classes of problems “between” PCA WITH OUTLIERS (most pessimistic perspective with the worst assumptions on the outliers) and ROBUST SUBSPACE RECOVERY (an optimistic perspective where all inliers belong to a subspace and all outliers do not).

As another example, consider the popular model where the inliers and the outliers are coming from a low-dimensional

[1] ETH of Impagliazzo, Paturi, and Zane (Impagliazzo et al., 2001) is that 3-SAT with n-variables is not solvable in time $2^{o(n)}$.

space plus some noise, and the noise of the outliers has heavy tail behavior. (See, e.g. distributional models in the line of work on Robust Probabilistic Principal Component Analysis starting with (Archambeau et al., 2006).) In such a scenario one can provably show that our assumptions will hold with good probability in the setting where d is large compared to r, and that the probability will improve with increasing d. In a way, the curse of dimensionality thus favors our assumptions.

For each of the α-assumptions, we provide an exact algorithm solving the corresponding version of PCA WITH OUTLIERS. The formal statement is given in Theorems 1.2 and 1.3 next.

Theorem 1.2. *For constant α, there exists a randomized algorithm for* PCA WITH OUTLIERS *that in time*

$$2^{\mathcal{O}(r(\log k+\log\log n)(r+\log n+\log(1/\delta)))}(n+d)^{\mathcal{O}(1)}$$

outputs a correct solution with success probability $1-\delta$ *under the α-gap assumption.*

Theorem 1.3. *For constant α, there exists a randomized algorithm for* PCA WITH OUTLIERS *that in time*

$$2^{\mathcal{O}(r^2(\log k+\log\log n)(\log k+\log\log n+\log(1/\delta)))}(n+d)^{\mathcal{O}(1)}$$

outputs a correct solution with success probability $1-\delta$ *under the α-heavy assumption.*

Observe that Theorem 1.3 provides a better running time than Theorem 1.2. In particular, it implies a fixed-parameter tractable (FPT) algorithm for PCA WITH OUTLIERS when the parameters are k, r, and the probability δ. That is, the running time is of the form $f(r,k,\delta)\cdot\mathrm{poly}(n,d)$ for a certain function f. However, the assumption of Theorem 1.3 is stronger.

The proofs of both theorems are based on the following strategy: apply randomized dimensionality reduction (sketching), and then use the methods of algebraic geometry to compute the exact solution. The difficulty is that in general, the dimensionality reduction distorts distances between the points. And at first thought such methods are not useful for obtaining *exact* solutions. The main technical contribution here is the proof that under α-gap and α-heavy assumptions, carefully designed sketches still can be used to obtain exact solutions. We believe that such sketches could find applications beyond robust PCA problems.

Finally, we prove two results about ROBUST SUBSPACE RECOVERY, the "simplest" variant of PCA WITH OUTLIERS. First, Theorem 1.4 gives an FPT algorithm with parameters k and r. Comparing with Theorems 1.3 and Theorem 1.2, the algorithm in Theorem 1.4 is faster for small values of k, and is deterministic.

Theorem 1.4. ROBUST SUBSPACE RECOVERY *is solvable in time* $2^{\mathcal{O}(k(\log r+\log k))}n^{\mathcal{O}(1)}$.

Second, we give the following hardness result. This is a conditional lower bound subject to the Exponential Time Hypothesis (ETH).

Theorem 1.5. *There is no algorithm solving* ROBUST SUBSPACE RECOVERY *in time $f(d)\cdot n^{o(d)}$ for any computable function f of d, unless ETH fails.*

In other words, it is hard even to verify if $\mathbf{A}$ could be represented as the sum of a low-rank matrix and an outlier matrix, without any error. From here, it easily follows that under the same assumption there is no algorithm for the general problem PCA WITH OUTLIERS giving *any* multiplicative approximation guarantee in time $f(d)\cdot n^{o(d)}$. This significantly extends the hardness result of (Simonov et al., 2019) which rules out only a constant-factor approximation. Also, since α-assumptions generalize the case of ROBUST SUBSPACE RECOVERY, the same hardness immediately holds for PCA WITH OUTLIERS with α-gap or α-heavy assumption. Observe that the running times in Theorems 1.2, 1.3 approach the lower bound of Theorem 1.5, in the following sense. By Theorem 1.5, at least the factor of $\mathcal{O}(r\log k)$ is required in the exponent, as, up to ETH, there is no $f(r)\cdot k^{o(r)}$-time algorithm, since $r\leq d$ and $k\leq n$. Due to the page limit, we defer the proofs of Theorem 1.4 and Theorem 1.5 to the supplementary material.

Related work. There is a vast amount of literature about robust PCA problems, see (Vaswani & Narayanamurthy, 2018; Xu et al., 2010; Bouwmans et al., 2016; Huber, 2004) for further references. There are two basic approaches to PCA with outliers in the literature. One is the development of information-theoretically optimal robust estimators, the fundamental issue of the robust statistics pioneered by Tukey and Huber in the 1960s, see (Huber, 2004; Tukey, 1977). For example, the famous least median of squares estimator of (Rousseeuw, 1984) for regression works even when half of the observations are outliers. Computationally the question of finding this estimator boils down to the problem of finding a d-dimensional subspace that minimizes the median Euclidean distance to the observation points. Only modest improvements over brute-force search trying all possible d-dimensional subspaces are known for resolving this question, see (Edelsbrunner & Souvaine, 1990). The second approach is to identify outliers by convex optimization. This approach, popularized by (Candès et al., 2011), (Wright et al., 2009), and (Chandrasekaran et al., 2011), has been centered around proving that, under some feasibility assumptions on the input, a convex relaxation of the initial optimization problem can recover the low-rank matrix. We refer to (Wright et al., 2009; Xu et al., 2010; Chen et al., 2011), and Chapter 3.3.2 in (Vidal et al., 2016) for further information on this approach. ROBUST SUBSPACE RECOVERY is also an extremely well-studied problem. For an overview of approaches based on convex and non-convex optimization methods, see (Maunu et al., 2019; Lerman & Maunu, 2018; Maunu & Lerman, 2019).

Much less is known about general-case algorithms with guaranteed performance for PCA WITH OUTLIERS and ROBUST SUBSPACE RECOVERY. For PCA WITH OUTLIERS, (Bhaskara & Kumar, 2018) provide two $(1+\varepsilon)$-approximation bicriteria algorithms. While the cost of their solution is preserved within $(1+\varepsilon)$ factor, the number of outliers k and the dimension of the subspace r in the solution are also approximated. Our Theorem 1.1 approximates only the cost of the solution. (Deshpande & Pratap, 2020) gave another $(1+\varepsilon)$-approximation for PCA WITH OUTLIERS in a more general ℓ_p-norm approximation. However, their algorithm works under the assumption that the ℓ_p error of the optimal subspace summed over the optimal inliers is at least δ times its total ℓ_p error summed over all n points, for some $\delta > 0$. In comparison, our Theorem 1.1 does not make any assumptions on the properties of the input, and the α-heavy and α-gap assumptions used in Theorems 1.2 and 1.3 are in a sense opposite to the assumption above: our assumptions hold in the case where outliers are sufficiently *far* from the optimal subspace, while for the result of (Deshpande & Pratap, 2020) the outliers must be sufficiently *close*.

The works of (Bhaskara & Kumar, 2018) and (Deshpande & Pratap, 2020) introduce some assumptions on the input structure and exploit these assumptions algorithmically. Comparing our α-gap and α-heavy assumptions with the rank-k condition number of (Bhaskara & Kumar, 2018), as well as the condition considered in (Deshpande & Pratap, 2020), our results are in a sense orthogonal to that. The rank-k condition number is small when the inliers are sufficiently far from the optimal low-dimensional subspace, while our α-gap and α-heavy assumptions hold in the opposite case, when the inliers are sufficiently close to the target subspace. In particular, α-gap and α-heavy assumptions cover the ROBUST SUBSPACE RECOVERY problem, while the rank-k condition and assumptions of (Deshpande & Pratap, 2020) do not.

(Hardt & Moitra, 2013) provides a polynomial time ROBUST SUBSPACE RECOVERY algorithm that works correctly when the number of inliers is at least $\frac{r}{d}n$ and inliers satisfy some linear independence property. Theorem 1.4 does not make any assumptions on the structure of inliers.

(Simonov et al., 2019) gave an algorithm that solves PCA WITH OUTLIERS in time roughly $n^{\mathcal{O}(r \cdot d)}$. Their algorithm is based on the methods of algebraic geometry. Theorems 1.2 and 1.3 provide much better running times, however the algorithms of (Simonov et al., 2019) does not require any assumptions on the properties of outliers. The proofs of Theorems 1.2 and 1.3 are based on a non-trivial adaptation of the technique introduced in (Simonov et al., 2019).

When it comes to lower bounds, Khachiyan in (Khachiyan, 1995) proved that it is NP-hard to find a $(d-1)$-dimensional subspace that contains at least $(1-\varepsilon)(1-1/d)n$ points. (Hardt & Moitra, 2013) used the Small Set Expansion conjecture (SSE) to show that for ROBUST SUBSPACE RECOVERY, even if one allows to select $(1+\delta)k$ outliers, for some $\delta > 0$, it is still unlikely that a polynomial time algorithm can find a $c \cdot r$-dimensional subspace containing all remaining $n-(1+\delta)k$ points for any $c > 0$. (Bhaskara & Kumar, 2018) used the smallest edge r-subgraph conjecture to show that, there is exists a constant $c > 0$, such that no polynomial time algorithm can find an rn^c-dimensional subspace that results in a multiplicative approximation to the objective cost. By using the hardness of the rank reduction problem for matroids, see Proposition 8.1 in (Fomin et al., 2018a), it is possible to show that ROBUST SUBSPACE RECOVERY is W[1]-hard parameterized by k. The lower bound of Theorem 1.5 is incomparable with these results.

(Simonov et al., 2019) proved a lower bound which (assuming ETH) rules out any constant-factor approximation of PCA WITH OUTLIERS in time $f(d) \cdot n^{o(d)}$, for any function f of d. Theorem 1.5 provides a stronger statement, it rules out *any* approximation for PCA WITH OUTLIERS within the same running time bound. Also, we believe our reduction is considerably simpler, and that it shows the connection between identifying subgraphs and identifying low-rank subsets more clearly.

2. Preliminaries

For a positive integer n, we use $[n]$ to denote the set $\{1, \ldots, n\}$. We use bold lowercase, e.g. $\mathbf{x} = [x_i]$, to denote a vector and bold uppercase, e.g. $\mathbf{A} = [a_{ij}]$, to denote a matrix. All our vectors are column vectors. The i-th row and the j-th column of $\mathbf{A}$ are denoted by $\mathbf{a}_{\mathbf{i}:}$ and $\mathbf{a}_{:\mathbf{j}}$ respectively. For $\mathbf{A} \in \mathbb{R}^{n \times d_1}$ and $\mathbf{B} \in \mathbb{R}^{n \times d_2}$, $[\mathbf{A}|\mathbf{B}]$ denotes the column-wise matrix concatenation of $\mathbf{A}$ and $\mathbf{B}$. The column and row spaces of a matrix are denoted by $\mathrm{col}(\mathbf{A})$ and $\mathrm{row}(\mathbf{A})$ respectively. For a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and an r-dimensional linear subspace $\mathcal{U}$ of $\mathbb{R}^d$ whose basis is given by rows of $\mathbf{U} \in \mathbb{R}^{r \times d}$, let $\mathrm{dist}^2(\mathbf{A}, \mathbf{U})$ (and $\mathrm{dist}^2(\mathbf{A}, \mathcal{U})$) represent sum of squares of ℓ_2-distances of rows of $\mathbf{A}$ from $\mathcal{U}$ which is equal to $\min_{\mathbf{X} \in \mathbb{R}^{n \times r}} \|\mathbf{A} - \mathbf{X}\mathbf{U}\|_F^2$. For non-negative real numbers a and b, we use the notation $a = (1 \pm \varepsilon)b$ if $a \in [(1-\varepsilon)b, (1+\varepsilon)b]$.

Subspace and affine embeddings. In our result we use dimensionality reduction tools from randomized numerical linear algebra literature. We refer the reader to the surveys of (Woodruff, 2014) and (Mahoney et al., 2011) for an overview of this area.

Definition 2.1 (ε-embedding)**.** *Given a subset $\mathcal{W} \subset \mathbb{R}^d$ and $\varepsilon \in (0, 1)$, an* ε-embedding *is a matrix $\mathbf{S} \in \mathbb{R}^{d \times s}$ for* *some $s \geq 0$ such that for all $\mathbf{x} \in \mathcal{W}$, we have*

$$\left\|\mathbf{x}^{\mathbf{T}}\mathbf{S}\right\|_2^2 = (1 \pm \varepsilon) \left\|\mathbf{x}\right\|_2^2 .$$

When $\mathcal{W}$ is a linear subspace we call $\mathbf{S}$ *an* ε-subspace embedding.

Essentially, an ε-embedding $\mathbf{S}$ is a linear transform, with small $s << d$, providing an approximate isometry over the embedded space $\mathcal{W} \subset \mathbb{R}^d$. In our work we will require embeddings for subspaces. To the best of our knowledge, the first usage of the notion of subspace embedding in numerical linear algebra was done in (Sarlos, 2006) and since then they have emerged as a powerful tool for accelerating various statistical learning procedure like ℓ_p-regression, low-rank approximation, and PCA. The theorem of (Indyk & Motwani, 1998) below provides the bound on embedding dimension s of a normal transform for it to be a subspace embedding.

Theorem 2.1 (**Normal Transform** (Indyk & Motwani, 1998))**.** *Let $0 < \varepsilon, \delta < 1$ and $\mathbf{S} = \frac{1}{\sqrt{s}}\mathbf{G} \in \mathbb{R}^{d\times s}$ where the entries of matrix* $\mathbf{G}$ *are independent standard normal random variables. Then if $s = \Theta((r + \log(1/\delta))\varepsilon^{-2})$, then for any fixed r-dimensional linear subspace $\mathcal{U} \subset \mathbb{R}^d$, with probability at least $1 - \delta$,* $\mathbf{S}$ *is an ε-subspace embedding.*

The following theorem, observed by (Sarlos, 2006), is an immediate application of ℓ_2-subspace embedding to ℓ_2-regression problem which says that the solution to embedded ℓ_2-regression problem provides a good approximate solution to the original regression problem. For completeness, we give the proof in the supplementary.

Theorem 2.2 ((Sarlos, 2006))**.** *Given a matrix $\mathbf{V} \in \mathbb{R}^{r\times d}$ and $\mathbf{a} \in \mathbb{R}^d$, we have*

$$\begin{aligned}(1 - \varepsilon)\operatorname{dist}^2(\mathbf{a}^{\mathbf{T}}, \mathbf{V}) &\leq \operatorname{dist}^2(\mathbf{a}^{\mathbf{T}}\mathbf{S}, \mathbf{VS}) \\ &\leq (1 + \varepsilon)\operatorname{dist}^2(\mathbf{a}^{\mathbf{T}}, \mathbf{V}).\end{aligned}$$

where $\mathbf{S} \in \mathbb{R}^{d\times s}$ is an ε-subspace embedding for subspace spanned by $\operatorname{row}(\mathbf{V})$ and $\mathbf{a}$.

Next, we recall the concept of *affine embeddings* from (Clarkson & Woodruff, 2013).

Definition 2.2. *Let $\mathbf{U} \in \mathbb{R}^{n\times r}$ and $\mathbf{B} \in \mathbb{R}^{n\times d'}$, then $\mathbf{S} \in \mathbb{R}^{s\times n}$ is an ε-affine embedding for $(\mathbf{U}, \mathbf{B})$ if for every $\mathbf{X} \in \mathbb{R}^{r\times d'}$, we have*

$$\left\|\mathbf{S}(\mathbf{UX} - \mathbf{B})\right\|_2^2 = (1 \pm \varepsilon) \left\|\mathbf{UX} - \mathbf{B}\right\|_2^2 .$$

(Clarkson & Woodruff, 2013) introduce several oblivious (data independent) constructions for affine embeddings like sparse embedding matrices, fast JL matrices, etc. These constructions vary in dimension they embed into, dependence of embedding dimension on the failure probability, and time it takes to apply them. For our results, we would need an oblivious construction with embedding dimension as small as possible and $\log(\frac{1}{\delta})$ dependence on the failure probability δ. The embedding dimension in various constructions of (Clarkson & Woodruff, 2013) are optimized while keeping the time taken to apply them small and with only $\frac{1}{\delta}$ dependence on the failure probability. For our purposes, the time taken to apply the sketch is not the bottleneck and thus we show the following theorem, which gives the optimal dependence on the embedding dimension. For the proof, see the supplementary.

Theorem 2.3. *Let $0 < \varepsilon, \delta < 1$ and $\mathbf{S} = \frac{1}{\sqrt{s}}\mathbf{G} \in \mathbb{R}^{s\times n}$, where the entries of matrix* $\mathbf{G}$ *are independent standard normal random variables. If $s = \mathcal{O}(r\log(1/\delta)\varepsilon^{-2})$, then for every fixed $\mathbf{U} \subset \mathbb{R}^{n\times r}$, with probability at least $1 - \delta$,* $\mathbf{S}$ *is an ε-affine embedding for* $\mathbf{U}$.

Sampling points from algebraic sets As a subroutine in our algorithms we use the fundamental results from algebraic geometry about sampling points from algebraic sets. See the book of (Basu et al., 2006) for further reference on the algorithmic algebraic geometry.

We denote the ring of polynomials in variables $X_1, \ldots, X_d$ with coefficients in $\mathbb{R}$ by $\mathbb{R}[X_1, \ldots, X_d]$. By saying that an algebraic set V in $\mathbb{R}^d$ is defined by $Q \in \mathbb{R}[X_1, \ldots, X_d]$, we mean that $V = \{x \in \mathbb{R}^d | Q(x_1, \ldots, x_d) = 0\}$. For a set of s polynomials $\mathcal{P} = \{P_1, \ldots, P_s\} \subset \mathbb{R}[X_1, \ldots, X_d]$, a sign condition at point $x \in V$ is defined as $\sigma_x^{\mathcal{P}} = (\operatorname{sign}(P_1(x)), \ldots, \operatorname{sign}(P_s(x)))$. An important question in real algebraic geometry is to compute a set of points realizing every possible sign condition as one ranges over V. The following theorem from (Basu et al., 2006) gives an algorithm to find such a set.

Proposition 2.4 ((Basu et al., 2006), Theorem 13.22)**.** *Let V be an algebraic set in $\mathbb{R}^d$ defined by $Q(X_1, \ldots, X_d) = 0$, where Q is a polynomial in $\mathbb{R}[X_1, \ldots, X_d]$ of degree at most b, and let $\mathcal{P} \subset \mathbb{R}[X_1, \ldots, X_d]$ be a finite set of s polynomials with each $P \in \mathcal{P}$ also of degree at most b. Let D be a ring generated by the coefficients of Q and the polynomials in $\mathcal{P}$. There is an algorithm which takes as input Q, d and $\mathcal{P}$ and computes a set of points realizing every possible sign condition in V over $\mathcal{P}$. The algorithm uses at most $(sb)^{O(d)}$ arithmetic operations in D.*

On the practical side, we note that a number of routines from (Basu et al., 2006) is implemented in the SARAG library (Caruso, 2006).

3. Approximation Scheme for PCA WITH OUTLIERS

We now outline the $n^{\mathcal{O}(r\log r\varepsilon^{-2})}$ time algorithm for solving the generic case of PCA WITH OUTLIERS with $(1 + \varepsilon)$-

factor approximation, as claimed by Theorem 1.1. The general idea is to observe that the unknown rows of $\mathbf{A}$ that form the inlier submatrix can be well approximated by a small-sized sample of them, in terms of low-rank approximation. This follows from established results saying that one can obtain a subspace embedding for a matrix by sampling and reweighting a few of its rows in proportion to a certain modification of their leverage scores. In particular, we employ the result of (Cohen et al., 2017) stating that sampling $\mathcal{O}(r \log r\varepsilon^{-2})$ rows in accordance with their ridge leverage scores provides a $(1+\varepsilon)$ approximation for any rank-r orthogonal projection.

The challenge here is that we do not know the actual inlier matrix to compute the scores and to perform the sampling from, as an arbitrary set of k rows of the given matrix might be outliers. However, we can guess the particular rows from a successful sample and also guess the approximated ridge leverage scores so that the optimal low-rank approximation of the resulting small matrix will also approximate well the unknown inlier matrix. Here we use crucially that constant-factor overestimates of the ridge leverage scores still suffice for the result of (Cohen et al., 2017). After this, it is only a matter of greedily selecting the rows of $\mathbf{A}$ that are the closest to the computed low-dimensional approximation space. The above summarizes the intuition behind Theorem 1.1, for the detailed proof we refer the reader to the supplementary material.

4. α-heavy and α-gap PCA WITH OUTLIERS

In this section we present Theorems 1.2 and 1.3, their proofs are done in two steps. First, we show an $2^{\mathcal{O}(\log k + \log\log n)rd}\,\mathrm{poly}(n,d,1/\delta)$ subspace-sampling algorithm for α-gap and α-heavy instances of PCA WITH OUTLIERS that succeeds with probability $1-\delta$, building upon the $n^{\mathcal{O}(rd)}$ algorithm from (Simonov et al., 2019). Second, we get rid of the exponential dependence on d by using dimensionality reduction techniques.

4.1. Subspace-sampling Algorithm

We start with briefly recalling the idea of the algorithm from (Simonov et al., 2019). One can parameterize the unknown r-dimensional subspace by $r \times d$ variables, and then for every pair of points construct a polynomial in these variables such that its sign determines the farthest point from a subspace. Thus the signs of all the $\binom{n}{2}$ polynomials determine exactly which k points are the farthest from the subspace, and so are the outliers. By enumerating all possible sign conditions via Proposition 2.4 in time $n^{\mathcal{O}(rd)}$, they get all potential sets of outliers.

The main idea for our algorithm is as follows. We show that it is possible to replace the trivial $\binom{n}{2}$-sized polynomial system by a much smaller one that still allows to detect the outliers, provided that either the α-gap or the α-heavy assumption holds. Intuitively, we first partition the points into $m = \Theta(k)$ buckets such that there is at most one outlier in each bucket w.h.p. Then, we compose $\binom{m}{2}$ polynomials to determine the k buckets containing the outliers, and for each bucket we also construct $\log n$ polynomials to detect the outlier in the bucket. Thus, our system contains only $\mathrm{poly}(k \log n)$ polynomials, providing the desired running time.

Now we give the proof in full detail. The algorithm, later denoted as Subspace-sampling algorithm, proceeds as follows:

1. Partition the rows of $\mathbf{A}$ (the points) into m buckets using perfect hashing. Let B_1, B_2,...,$B_m \subset [n]$ be the indices of points in each bucket.

2. For each bucket B_i, $i \in [m]$, construct the set of polynomials $\mathcal{P}_i = \{P_i^j\}_{1 \le j \le \log_2 n}$,

   $$P_i^j(\mathbf{V}) = \sum_{\substack{\ell \in B_i \\ \ell_j = 1}} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}))^b - \sum_{\substack{\ell \in B_i \\ \ell_j = 0}} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}))^b$$

   where ℓ_j is the j-th bit in the binary representation of ℓ and $b = \Theta(\frac{\log n}{\log(1+\alpha)})$ for α-gap instances and $b = 1$ for α-heavy instances. Note that $\mathbf{V}$ can be parameterized by $r \times d$ variables such that for each $\ell \in [n]$, $\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V})$ can be expressed as a constant-degree polynomial in these variables. See (Simonov et al., 2019) for details.

3. Also consider the set of polynomials $\mathcal{Q} = \{Q_{i,j}\}_{1 \le i < j \le m}$, where

   $$Q_{i,j}(\mathbf{V}) = \sum_{\ell \in B_i} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}))^b - \sum_{\ell \in B_j} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}))^b$$

   and let $\mathcal{P}$ be the collection of all the polynomials defined above, i.e., $\mathcal{P} = \bigcup_i \mathcal{P}_i \cup \mathcal{Q}$.

4. Using Proposition 2.4, enumerate all possible sign conditions of $\mathcal{P}$ on $\mathcal{X}$, the space of all r-dimensional subspaces, i.e. compute the set $\mathcal{T} = \mathrm{Sample}(\mathbf{A}, r, k)$ where

   $$\mathrm{Sample}(\mathbf{A}, r, k) = \{(S, \mathbf{V}_S) | S \text{ is a sign condition of } \mathcal{P} \text{ on } \mathcal{X} \text{ and } \mathbf{V}_S \text{ realizes } S\}$$

5. For each $(S, \mathbf{V}_S) \in \mathcal{T}$, do the following:

   (a) Note that the signs of $\mathcal{Q}$ give an ordering on $[m]$. Take the top k indices from this ordering and let w.l.o.g. this set be $[k]$.

(b) For each B_i, $i \in [k]$, choose $p \in B_i$ such that p_j is set to 1 if $P_i^j(\mathbf{V}_S) > 0$ and 0 otherwise. This gives us k rows of $\mathbf{A}$, one from each bucket, let $\mathbf{N} \in \mathbb{R}^{n \times d}$ be the matrix containing these k rows at their respective positions in $\mathbf{A}$.

(c) Find the optimal r-dimensional projection $\mathbf{L}$ of $\mathbf{A} - \mathbf{N}$ via the vanilla PCA algorithm.

6. Return $\mathbf{N}$ and $\mathbf{L}$ from step 5 with the minimum cost of projection.

Running time. The sampling step 4 dominates the running time. Since $|\mathcal{P}| = m \log n + \binom{m}{2} = O(k \log n + k^2)$ and the degree of all the polynomials involved in $\mathcal{P}$ is bounded by $\mathcal{O}(\frac{\log n}{\log(1+\alpha)})$, Proposition 2.4 gives the time $2^{\mathcal{O}(\log k + \log\log n - \log\log(1+\alpha))rd}$, and this bound also holds for the size of $\mathcal{T}$. Also in step 1, in time $\mathcal{O}(n/\delta)$ one can construct a perfect hash function with success probability $1 - \delta$. All the other steps of the algorithm take $\mathrm{poly}(n, d)$ time. So the total running time is $2^{\mathcal{O}(\log k + \log\log n - \log\log(1+\alpha))rd}\,\mathrm{poly}(n, d, 1/\delta)$.

Correctness of the algorithm. Assume that in the optimal solution the outlier matrix is $\mathbf{N}^*$, and the low-rank matrix is $\mathbf{L}^*$. Denote by $\mathbf{V}^*$ the r-dimensional subspace corresponding to $\mathbf{L}^*$. The following claim shows that the sign condition corresponding to $\mathbf{V}^*$ allows the algorithm to restore $\mathbf{N}^*$.

Claim 1. *In step 5 of the algorithm, with high probability, the outlier matrix $\mathbf{N}$ generated on the sign condition S^* which comes from evaluating $\mathcal{P}$ on $\mathbf{V}^*$ is equal to $\mathbf{N}^*$.*

Proof. **Getting buckets with outliers, Step 5(a)**: Note that in Step 1 of the algorithm we map points to $\mathcal{O}(k)$ buckets in order to ensure that each bucket has at most one outlier. So some buckets will end up having no outliers at all. We show that in step 5(a) of the algorithm for the sign condition S^* we correctly find the k buckets containing outlier points. Specifically, for a bucket B_i with an outlier point p and a bucket B_j with no outliers we show that $Q_{i,j}(\mathbf{V}^*) > 0$. For α-gap instances we have

$$
\begin{aligned}
&Q_{i,j}(\mathbf{V}^*) = \sum_{\ell \in B_i} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*))^b - \sum_{\ell \in B_j} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*))^b \\
&\geq (\mathrm{dist}^2(\mathbf{a}_{\mathbf{p}:}^{\mathbf{T}}, \mathbf{V}^*))^b - n \max_{\ell \in I} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*))^b \\
&\geq ((1+\alpha)^b - n) \max_{\ell \in I} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*))^b > 0.
\end{aligned}
$$

Similarly, for α-heavy instances we have

$$
\begin{aligned}
Q_{i,j}(\mathbf{V}^*) &= \sum_{\ell \in B_i} \mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*) - \sum_{\ell \in B_j} \mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*) \\
&\geq \mathrm{dist}^2(\mathbf{a}_{\mathbf{p}:}^{\mathbf{T}}, \mathbf{V}^*) - \sum_{\ell \in I} \mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*) \\
&\geq \alpha \sum_{\ell \in I} \mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*) > 0.
\end{aligned}
$$

Therefore, taking the top k buckets by the ordering induced by the signs of $\mathcal{Q}(\mathbf{V}^*)$ will give us precisely the k buckets containing the outlier points. WLOG, let $B_1, B_2, ..., B_k$ be those buckets, later referred to as the outlier buckets.

Extracting outliers from the outlier buckets, Step 5(b): Fix an $i \in [k]$, denote the sole outlier point in the bucket B_i by p. We show that for each $j \in [\log n]$, $p_j = 1$ iff $P_i^j(\mathbf{V}^*) > 0$ and $p_j = 0$ iff $P_i^j(\mathbf{V}^*) < 0$. For α-gap instances, if $p_j = 1$ then we have

$$
\begin{aligned}
P_i^j(\mathbf{V}) &= \sum_{\substack{\ell \in B_i \\ \ell_j = 1}} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}))^b - \sum_{\substack{\ell \in B_i \\ \ell_j = 0}} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}))^b \\
&\geq (\mathrm{dist}^2(\mathbf{a}_{\mathbf{p}:}^{\mathbf{T}}, \mathbf{V}^*))^b - n \max_{\ell \in I} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*))^b \\
&\geq ((1+\alpha)^b - n) \max_{\ell \in I} (\mathrm{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*))^b \\
&> 0.
\end{aligned}
$$

Analogously, if $p_j = 0$, then $P_i^j(\mathbf{V}^*) < 0$. Similar analysis works for α-heavy instances with $b = 1$.

Thus the outlier matrix $\mathbf{N}$ generated on the sign condition S^* in step 5 is precisely the optimal $\mathbf{N}^*$. Success of the algorithm relies on perfect hashing of the outlier points into m buckets in step 1 and for $m = \Theta(k)$ one can find in time $\mathcal{O}(n/\delta)$ a perfect hash function with success probability $1 - \delta$. □

Since in our algorithm we go over all possible sign conditions of $\mathcal{P}$ on $\mathcal{X}$, the sign condition S^* will also be considered, and will provide the optimal outlier matrix. Once we have the optimal $\mathbf{N}^*$, computing the optimal rank-r projection of $\mathbf{A} - \mathbf{N}^*$ as the matrix $\mathbf{L}$ will give us the optimal cost. Thus, Claim 1 implies the correctness of the algorithm.

4.2. Dimensionality Reduction

In this subsection we improve upon the d in the exponent of the running time of the algorithm from the previous section. We achieve this by observing that for α-gap and α-heavy instances we only need the approximate distances of points to a subspace instead of the exact ones. The new algorithm proceed as follows:

1. Sample a normal transform matrix $\mathbf{S} \in \mathbb{R}^{d \times t}$, where $t = \mathcal{O}(r + \log n + \log(1/\delta))$ for α-gap instances and $t = \mathcal{O}(r(\log k + \log\log n + \log(1/\delta))$ for α-heavy instances.

2. Sketch the input matrix $\mathbf{A}$ from the right, $\widetilde{\mathbf{A}} = \mathbf{AS}$.

3. Find the optimal set of outliers for $\widetilde{\mathbf{A}}$ using the algorithm from the previous subsection.

4. Construct the matrix $\mathbf{N}$ from the corresponding rows of $\mathbf{A}$, and return $\mathbf{N}$ together with the optimal rank-r projection $\mathbf{L}$ of $\mathbf{A} - \mathbf{N}$.

Running Time. Clearly, step 3 dominates the running time. Since the ambient dimension is reduced from d to t, the runtime of the new algorithm is $2^{O(r(\log k+\log\log n)(r+\log n+\log(1/\delta)))}\operatorname{poly}(n, d)$ for α-gap instances and $2^{\mathcal{O}(r^2(\log k+\log\log n)(\log k+\log\log n+\log(1/\delta))}\operatorname{poly}(n, d)$ for α-heavy instances.

Correctness of the algorithm. Correctness of the algorithm relies on the following two lemmas, handling α-gap instances and α-heavy instances respectively. Intuitively, we prove that a suitable embedding preserves the set of the optimal outliers.

Lemma 4.1. *Let* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *and integer parameters* r *and* k *be an* α*-gap instance of* PCA WITH OUTLIERS. *Let* $\mathbf{S} \in \mathbb{R}^{d\times t}$ *be an* ε*-embedding for*

$$\operatorname{col}([\mathbf{V}^{*\mathbf{T}}|\mathbf{a_{1:}}]), \operatorname{col}([\mathbf{V}^{*\mathbf{T}}|\mathbf{a_{2:}}]), \ldots, \operatorname{col}([\mathbf{V}^{*\mathbf{T}}|\mathbf{a_{n:}}]),$$

simultaneously, for a small enough constant ε*. Here* $\mathbf{V}^* \in \mathbb{R}^{r\times d}$ *is the* r*-dimensional linear subspace which spans the rows of the optimal rank-*r *matrix* $\mathbf{L}^*$*. Let* $\widetilde{\mathbf{A}} = \mathbf{AS}$ *and* $\widetilde{\mathcal{T}} = \operatorname{Sample}(\widetilde{\mathbf{A}}, r, k)$ *where* Sample *is the procedure from step 4 of the Subspace-sampling algorithm. Then there exists* $(\widetilde{C}, \widetilde{\mathbf{U}}) \in \widetilde{\mathcal{T}}$ *such that the outlier matrix* $\widetilde{\mathbf{N}}$ *generated on* $(\widetilde{C}, \widetilde{\mathbf{U}})$ *in step 5 of the Subspace-sampling algorithm is same as the optimal outlier matrix* $\mathbf{N}^*$

Proof. We begin by observing that the distances of the rows of $\mathbf{AS}$ from $\operatorname{row}(\mathbf{V}^*\mathbf{S})$ are the same as the distances of rows of $\mathbf{A}$ from $\operatorname{row}(\mathbf{V}^*)$, up to a constant factor distortion. Since $\mathbf{S}$ is a ε-embedding for $\operatorname{col}([\mathbf{V}^{*\mathbf{T}}|\mathbf{a_{1:}}]), \operatorname{col}([\mathbf{V}^{*\mathbf{T}}|\mathbf{a_{2:}}]), \ldots, \operatorname{col}([\mathbf{V}^{*\mathbf{T}}|\mathbf{a_{n:}}])$ simultaneously, using Theorem 2.2 we have for each $i \in [n]$

$$\begin{aligned}(1-\varepsilon)\operatorname{dist}^2(\mathbf{a}_{\mathbf{i:}}^{\mathbf{T}}, \mathbf{V}^*) &\le \operatorname{dist}^2(\mathbf{a}_{\mathbf{i:}}^{\mathbf{T}}\mathbf{S}, \mathbf{V}^*\mathbf{S}) \\ &\le (1+\varepsilon)\operatorname{dist}^2(\mathbf{a}_{\mathbf{i:}}^{\mathbf{T}}, \mathbf{V}^*).\end{aligned} \tag{1}$$

Now let $\widetilde{\mathcal{P}} = \cup\widetilde{\mathcal{P}^i}\cup\widetilde{\mathcal{Q}}$ be the collection of polynomials same as $\mathcal{P}$, but defined on the smaller space, i.e. the space of all r-dimensional subspaces in $\mathbb{R}^t$, with $\mathbf{AS}$ as the input matrix. Let $(\widetilde{C}, \mathbf{V}^*\mathbf{S}) \in \widetilde{T}$ be the sign condition of $\widetilde{\mathcal{P}}$ on $\mathbf{V}^*\mathbf{S}$. We claim that the outlier matrix, $\mathbf{N}$, generated on $(\widetilde{C}, \mathbf{V}^*\mathbf{S})$ in the step 5 of the Subspace-sampling algorithm is the optimal outlier matrix $\mathbf{N}^*$. To see this, first we we claim that in step 5(a) the top k indices obtained from ordering on $[m]$ given by signs of $\mathbf{V}^*\mathbf{S}$ on $\widetilde{\mathcal{Q}}$ give us k buckets containing outlier points. Note that to prove this it suffices to show that $\widetilde{Q_{i,j}}(\mathbf{V}^*\mathbf{S}) > 0$ for a bucket B_i with an outlier point p and a bucket B_j with no outlier point. Starting with the definition of $\widetilde{Q_{i,j}}(\mathbf{V}^*\mathbf{S})$,

$$\begin{aligned}&\sum_{\ell\in B_i}(\operatorname{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}\mathbf{S}, \mathbf{V}^*\mathbf{S}))^b - \sum_{\ell\in B_j}(\operatorname{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}\mathbf{S}, \mathbf{V}^*\mathbf{S}))^b \\ &\ge \sum_{\ell\in B_i}((1-\varepsilon)\operatorname{dist}^2(\mathbf{a}_{\ell:}, \mathbf{V}^*))^b - \sum_{\ell\in B_j}((1+\varepsilon)\operatorname{dist}^2(\mathbf{a}_{\ell:}, \mathbf{V}^*))^b \\ &\ge (1-\varepsilon)^b(\operatorname{dist}^2(\mathbf{a}_{\mathbf{p}:}, \mathbf{V}^*))^b - (1+\varepsilon)^b n \max_{\ell\in I}(\operatorname{dist}^2(\mathbf{a}_{\ell:}, \mathbf{V}^*))^b \\ &\ge ((1-\varepsilon)^b(1+\alpha)^b - n(1+\varepsilon)^b)\max_{\ell\in I}(\operatorname{dist}^2(\mathbf{a}_{\ell:}, \mathbf{V}^*))^b \\ &> 0 \quad \text{(For an appropriate } \varepsilon\text{)}\end{aligned}$$

where we have used (1) in the first inequality and the α-gap property in the third inequality.

Similarly the signs of $\widetilde{\mathcal{P}}$ on $\mathbf{V}^*\mathbf{S}$ in the Subspace-sampling algorithm will be able to retrieve the outlier points $\mathbf{N}^*$. The analysis is analogous to the above. □

Lemma 4.2. *Let* $\mathbf{A} \in \mathbb{R}^{n\times d}$ *and integer parameters* r *and* k *be a* α*-heavy instance of* PCA WITH OUTLIERS. *After bucketing points in step* 1 *of the Subspace-sampling algorithm let* $I_{i,j}^1$ *and* $I_{i,j}^0$ *be sets of indices of points in bucket* B_i *whose* j*-bit is* 1 *and* 0 *respectively. Let* $\mathbf{S} \in \mathbb{R}^{d\times t}$ *be a* ε*-affine embedding for* $\{(\mathbf{V}^{*T}, \mathbf{A}[I_{i,j}^k\ :]^T)\}_{\substack{1\le i\le m\\ 1\le j\le \log n\\ 0\le k\le 1}}$ *simultaneously, where* ε *is a sufficiently small constant. Here* $\mathbf{V}^* \in \mathbb{R}^{r\times d}$ *is the* r*-dimensional linear subspace which spans the rows of optimal* $\mathbf{L}$*. Let* $\widetilde{\mathbf{A}} = \mathbf{AS}$ *and* $\widetilde{\mathcal{T}} = \operatorname{Sample}(\widetilde{\mathbf{A}}, r, k)$ *where* Sample *is the procedure from step 4 of the Subspace-sampling algorithm. Then there exist* $(\widetilde{C}, \widetilde{\mathbf{U}}) \in \widetilde{\mathcal{T}}$ *such that the outlier matrix* $\widetilde{\mathbf{N}}$ *generated on* $(\widetilde{C}, \widetilde{\mathbf{U}})$ *in step 5 of the Subspace-sampling algorithm is same as the optimal outlier matrix* $\mathbf{N}^*$

Proof. We start by observing that since $\mathbf{S}$ is a ε-affine embedding for $\{\mathbf{V}^{*T}, \mathbf{A}[I_{i,j}^k\ :]^T\}_{\substack{1\le i\le m\\ 1\le j\le \log n\\ 0\le k\le 1}}$ simultaneously, we have that for all $i \in [m]$, $j \in [\log n]$ and $k \in \{0, 1\}$

$$\begin{aligned}(1-\varepsilon)\sum_{\substack{\ell\in B_i\\ \ell_j=k}}\operatorname{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*) &\le \sum_{\substack{\ell\in B_i\\ \ell_j=k}}\operatorname{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}\mathbf{S}, \mathbf{V}^*\mathbf{S}) \\ &\le (1+\varepsilon)\sum_{\substack{\ell\in B_i\\ \ell_j=k}}\operatorname{dist}^2(\mathbf{a}_{\ell:}^{\mathbf{T}}, \mathbf{V}^*).\end{aligned} \tag{2}$$

Now let $\widetilde{\mathcal{P}} = \cup\widetilde{\mathcal{P}^i}\cup\widetilde{\mathcal{Q}}$ be the collection of polynomials same as $\mathcal{P}$ but defined on smaller space, i.e. all r-dimensional subspaces of t-dimensional space, with $\mathbf{AS}$ as the input matrix. Let $(\widetilde{C}, \mathbf{V}^*\mathbf{S}) \in \widetilde{T}$ be the sign condition of $\widetilde{\mathcal{P}}$ on $\mathbf{V}^*\mathbf{S}$. We claim that the outlier matrix, $\mathbf{N}$, generated on $(\widetilde{C}, \mathbf{V}^*\mathbf{S})$ in the step 5 of the Subspace-sampling algorithm is the optimal outlier matrix $\mathbf{N}^*$. To see this note that that in step 5(a) the top k indices obtained from ordering on $[s]$ given by signs of $\mathbf{V}^*\mathbf{S}$ on $\widetilde{\mathcal{Q}}$ gives us k-buckets with outlier points. To prove this it suffices to show that $\widetilde{Q_{i,j}}(\mathbf{V}^*\mathbf{S}) > 0$ for a bucket B_i with an outlier point p and a bucket B_j with no outlier point. Starting with the definition of $\widetilde{Q_{i,j}}(\mathbf{V}^*\mathbf{S})$,

$$\begin{aligned}
&\sum_{\ell \in B_i} (\mathrm{dist}^2(\mathbf{a}_{\ell:}\mathbf{S}, \mathbf{V}^*\mathbf{S})) - \sum_{\ell \in B_j} (\mathrm{dist}^2(\mathbf{a}_{\ell:}\mathbf{S}, \mathbf{V}^*\mathbf{S})) \\
&\geq (1-\varepsilon) \sum_{\ell \in B_i} (\mathrm{dist}^2(\mathbf{a}_{\ell:}, \mathbf{V}^*)) - (1+\varepsilon) \sum_{\ell \in B_j} (\mathrm{dist}^2(\mathbf{a}_{\ell:}, \mathbf{V}^*)) \\
&\geq (1-\varepsilon)(\mathrm{dist}^2(\mathbf{a}_{\mathbf{p}:}, \mathbf{V}^*)) - (1+\varepsilon) \sum_{\ell \in I} (\mathrm{dist}^2(\mathbf{a}_{\ell:}, \mathbf{V}^*)) \\
&\geq ((1-\varepsilon)(1+\alpha) - (1+\varepsilon)) \sum_{\ell \in I} (\mathrm{dist}^2(\mathbf{a}_{\ell:}, \mathbf{V}^*)) \\
&> 0 \quad \text{(For appropriate } \varepsilon\text{)}
\end{aligned}$$

where we have used (2) in the first inequality and the α-heavy property in the third inequality.

Next we claim that using signs of $\widetilde{\mathcal{P}}$ on $\mathbf{V}^*\mathbf{S}$ the Subspace-sampling algorithm will be able to retrieve the outlier points $\mathbf{N}^*$. Analysis is analogous to the above. □

Lemma 4.1 and 4.2 prove the correctness of the algorithm, given that the sketching matrix $\mathbf{S}$ satisfies the conditions in the lemmas. Next we prove that the embedding dimension t is large enough for that to happen with probability at least $1-\delta$.

Claim 2. *Let $\varepsilon, \delta \in (0,1)$ and $\mathbf{S} = \frac{1}{\sqrt{s}}\mathbf{G} \in \mathbb{R}^{d \times s}$ where the entries of $\mathbf{G}$ are independent standard normal random variables with $s = \mathcal{O}((r + p + \log(1/\delta))\varepsilon^{-2})$ and $p = \min(\mathrm{rank}(\mathbf{A}), \log n)$. Then $\mathbf{S}$ is a ε-embedding simultaneously for $\mathrm{col}([\mathbf{U}^\mathbf{T}|\mathbf{a}_{\mathbf{1}:}])$, $\mathrm{col}([\mathbf{U}^\mathbf{T}|\mathbf{a}_{\mathbf{2}:}]), \ldots, \mathrm{col}([\mathbf{U}^\mathbf{T}|\mathbf{a}_{\mathbf{n}:}])$, for fixed $\mathbf{U} \in \mathbb{R}^{r \times d}$, with probability at least $1-\delta$.*

Proof. Consider the following two arguments.

1. By Theorem 2.1, for a fixed $i \in \{1, \ldots, n\}$, $\mathbf{S}$ is an ε-subspace embedding for $\mathrm{col}([\mathbf{U}^\mathbf{T}|\mathbf{a}_{\mathbf{i}:}])$ with probability at least $1 - \frac{\delta}{2^p}$. By the union bound, $\mathbf{S}$ is the desired ε-embedding with probability at least $1 - \frac{\delta}{2^p}n$.

2. Let $\mathcal{B} = \{\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b}_{\mathrm{rank}(\mathbf{A})}\}$ be the row basis for $\mathbf{A}$ and $\mathcal{V}' = \mathrm{col}([\mathbf{U}^\mathbf{T}|\mathbf{b_1}|\mathbf{b_2}|\cdots|\mathbf{b}_{\mathrm{rank}(\mathbf{A})}])$. Since $\mathrm{col}([\mathbf{U}^\mathbf{T}|\mathbf{a}_{\mathbf{i}:}]) \subseteq \mathcal{V}'$ for each $i \in [n]$, and $\dim(\mathcal{V}') \leq r + \mathrm{rank}(\mathbf{A})$, by Theorem 2.1, we have that $\mathbf{S}$ is an ε-embedding for $\mathcal{V}$ with probability at least $1 - \frac{\delta}{2^p}2^{\mathrm{rank}(\mathbf{A})}$.

Combining the above two arguments we have that $\mathbf{S}$ is ε-embedding for each $\mathrm{col}([\mathbf{U}^\mathbf{T}|\mathbf{a}_{\mathbf{i}:}])$, $i \in [n]$, with probability at least $1-\delta$. □

Claim 3. *Let $\varepsilon, \delta \in (0,1)$ and $\mathbf{S} = \frac{1}{\sqrt{t}}\mathbf{G} \in \mathbb{R}^{d \times t}$ where the entries of $\mathbf{G}$ are independent standard normal random variables with $t = \Theta(r(\log k + \log\log n + \log(1/\delta))\varepsilon^{-2})$. Then $\mathbf{S}$ is a ε-affine embedding simultaneously for $\{\mathbf{V}^{*T}, \mathbf{A}[I_{i,j}^k :]^T\}_{\substack{1 \leq i \leq m \\ 1 \leq j \leq \log n \\ 0 \leq k \leq 1}}$, for fixed $\mathbf{V} \in \mathbb{R}^{r \times d}$, with probability $1-\delta$.*

Proof. Follows from Theorem 2.3 and union bound. □

Finally, we observe that we have two sources of error in our algorithm. One is coming from the Subspace-sampling algorithm and the other from the dimensionality reduction. Setting failure probability to $\delta/2$ in each of them and applying union bound gives us $1-\delta$ success probability.

Acknowledgements

This work is supported by the Research Council of Norway via the project “MULTIVAL”.

References

Archambeau, C., Delannay, N., and Verleysen, M. Robust probabilistic projections. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML ’06, pp. 33–40, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143849. URL https://doi.org/10.1145/1143844.1143849.

Basu, S., Pollack, R., and Roy, M.-F. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 3540330984.

Bhaskara, A. and Kumar, S. Low rank approximation in the presence of outliers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, volume 116 of *LIPIcs*, pp. 4:1–4:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi: 10.4230/LIPIcs.APPROX-RANDOM.2018.4. URL https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2018.4.

Bouwmans, T., Aybat, N. S., and Zahzah, E.-h. *Handbook of robust low-rank and sparse matrix decomposition: Applications in image and video processing*. Chapman and Hall/CRC, 2016.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, 2011. doi: 10.1145/1970392.1970395. URL http://doi.acm.org/10.1145/1970392.1970395.

Caruso, F. The SARAG library: Some algorithms in real algebraic geometry. In Iglesias, A. and Takayama, N.

(eds.), *Mathematical Software - ICMS 2006*, pp. 122–131, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-38086-3.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011. doi: 10.1137/090761793. URL https://doi.org/10.1137/090761793.

Chen, J., Huang, X., Kanj, I. A., and Xia, G. Strong computational lower bounds via parameterized complexity. *J. Computer and System Sciences*, 72(8):1346–1367, 2006.

Chen, Y., Xu, H., Caramanis, C., and Sanghavi, S. Robust matrix completion and corrupted columns. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 873–880, 2011.

Clarkson, K. L. and Woodruff, D. P. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 81–90. ACM, 2013.

Cohen, M. B., Musco, C., and Musco, C. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1758–1777, USA, 2017. SIAM.

Croux, C. and Haesbroeck, G. Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87(3):603–618, 09 2000. ISSN 0006-3444. doi: 10.1093/biomet/87.3.603. URL https://doi.org/10.1093/biomet/87.3.603.

Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., and Saurabh, S. *Parameterized Algorithms*. Springer, 2015. URL http://dx.doi.org/10.1007/978-3-319-21275-3.

Deshpande, A. and Pratap, R. Subspace approximation with outliers. *CoRR*, abs/2006.16573, 2020. URL https://arxiv.org/abs/2006.16573.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Edelsbrunner, H. and Souvaine, D. L. Computing least median of squares regression lines and guided topological sweep. *Journal of the American Statistical Association*, 85(409):115–119, 1990.

Fomin, F. V., Golovach, P. A., Lokshtanov, D., and Saurabh, S. Covering vectors by spaces: Regular matroids. *SIAM J. Discret. Math.*, 32(4):2512–2565, 2018a. doi: 10.1137/17M1151250. URL https://doi.org/10.1137/17M1151250.

Fomin, F. V., Lokshtanov, D., Meesum, S. M., Saurabh, S., and Zehavi, M. Matrix rigidity from the viewpoint of parameterized complexity. *SIAM J. Discrete Math.*, 32(2):966–985, 2018b. doi: 10.1137/17M112258X. URL https://doi.org/10.1137/17M112258X.

Hanson, D. L. and Wright, F. T. A bound on tail probabilities for quadratic forms in independent random variables. *The Annals of Mathematical Statistics*, 42(3):1079–1083, 1971.

Hardt, M. and Moitra, A. Algorithms and hardness for robust subspace recovery. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, volume 30 of *JMLR Proceedings*, pp. 354–375. JMLR.org, 2013.

Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

Huber, P. J. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

Impagliazzo, R., Paturi, R., and Zane, F. Which problems have strongly exponential complexity. *J. Computer and System Sciences*, 63(4):512–530, 2001.

Indyk, P. and Motwani, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 604–613. ACM, 1998.

Kane, D. M. and Nelson, J. Sparser johnson-lindenstrauss transforms. *J. ACM*, 61(1):1–23, 2014.

Khachiyan, L. On the complexity of approximating extremal determinants in matrices. *J. Complex.*, 11(1):138–153, 1995. doi: 10.1006/jcom.1995.1005. URL https://doi.org/10.1006/jcom.1995.1005.

Lerman, G. and Maunu, T. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8):1380–1410, 2018.

Mahoney, M. W. et al. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

Maunu, T. and Lerman, G. Robust subspace recovery with adversarial outliers. *CoRR*, abs/1904.03275, 2019. URL http://arxiv.org/abs/1904.03275.

Maunu, T., Zhang, T., and Lerman, G. A well-tempered landscape for non-convex robust subspace recovery. *J. Mach. Learn. Res.*, 20:37:1–37:59, 2019. URL http://jmlr.org/papers/v20/17-324.html.

Pearson, K. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11): 559–572, 1901.

Roughgarden, T. Introduction. In *Beyond the Worst-Case Analysis of Algorithms*, pp. 1–24. 2020. doi: 10.1017/9781108637435.002. URL https://doi.org/10.1017/9781108637435.002.

Rousseeuw, P. J. Least median of squares regression. *J. Amer. Statist. Assoc.*, 79(388):871–880, 1984. ISSN 0162-1459. URL http://links.jstor.org/sici?sici=0162-1459(198412)79:388<871:LMOSR>2.0.CO;2-K&origin=MSN.

Sarlos, T. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 143–152, 2006.

Simonov, K., Fomin, F., Golovach, P., and Panolan, F. Refined complexity of PCA with outliers. In *In Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 5818–5826, 2019.

Tukey, J. W. *Exploratory data analysis*, volume 2. Reading, Mass., 1977.

Vaswani, N. and Narayanamurthy, P. Static and dynamic robust PCA and matrix completion: A review. *Proceedings of the IEEE*, 106(8):1359–1379, 2018. doi: 10.1109/JPROC.2018.2844126. URL https://doi.org/10.1109/JPROC.2018.2844126.

Vidal, R., Ma, Y., and Sastry, S. S. *Generalized Principal Component Analysis*, volume 40 of *Interdisciplinary applied mathematics*. Springer, 2016. ISBN 978-0-387-87810-2. doi: 10.1007/978-0-387-87811-9. URL https://doi.org/10.1007/978-0-387-87811-9.

Woodruff, D. P. *Sketching as a Tool for Numerical Linear Algebra*, volume 10 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers Inc., 2014.

Wright, J., Ganesh, A., Rao, S. R., Peng, Y., and Ma, Y. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Proceedings of 23rd Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 2080–2088. Curran Associates, Inc., 2009.

Xu, H., Caramanis, C., and Sanghavi, S. Robust PCA via outlier pursuit. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 2496–2504. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/4005-robust-pca-via-outlier-pursuit.