Unlabeled Principal Component Analysis

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce robust principal component analysis from a data matrix in which the entries of its columns have been corrupted by permutations, termed Unlabeled Principal Component Analysis (UPCA). Using algebraic geometry, we establish that UPCA is a well-defined algebraic problem in the sense that the only matrices of minimal rank that agree with the given data are row-permutations of the ground-truth matrix, arising as the unique solutions of a polynomial system of equations. Further, we propose an efficient two-stage algorithmic pipeline for UPCA suitable for the practically relevant case where only a fraction of the data have been permuted. Stage-I employs outlier-robust PCA methods to estimate the ground-truth column-space. Equipped with the column-space, Stage-II applies recent methods for unlabeled sensing to restore the permuted data. Experiments on synthetic data, face images, educational and medical records reveal the potential of UPCA for applications such as data privatization and record linkage.

1 Introduction

Motivation. In principal component analysis, a cornerstone of machine learning and data science, one is given a data matrix $\tilde{X}$, assumed to be a corrupted version of a ground-truth data matrix $X^* = [x_1^* \cdots x_n^*] \in \mathbb{R}^{m\times n}$, typically but not necessarilly assumed to have low rank, and the objective is to estimate X^* or the column-space $S^* \subset \mathbb{R}^m$ of X^*. The most common types of corruptions that have attracted interest in modern studies are additive sparse perturbations [6, 43], outlier data points that lie away from S^* [40, 37] and missing entries, the latter also known as matrix completion [7, 4].

Recently, permutations have been emerging as another type of data corruption, typically set in the context of linear regression, where the correspondences between the input and the output data have been partially distorted or even are entirely unavailable [35, 36, 21, 16, 28, 29, 42, 18, 38, 33]. There, one is given a point x^* of a linear subspace S^*, but only up to a permutation of its coordinates, say $\tilde{x} = \Pi^* x^*$ with Π^* an unknown permutation, and the goal is to find x^* from the data $\tilde{x}, S^*$. This *Unlabeled Sensing* [35, 36] problem has many potential machine learning applications, e.g., record linkage [28, 29], visual [24, 25] or textual [5, 26, 27] permutation learning, and matching problems in neuroscience [20] and biology [1, 39].

While methods for unlabeled sensing rely on knowledge of the source subspace S^*, this is not always known in practice. On the other hand, data of the form $\tilde{X} = [\tilde{x}_1, \ldots, \tilde{x}_n] \in \mathbb{R}^{m\times n}$ with $\tilde{x}_j = \Pi_j^* x_j^*$ an unknown permutation of an unknown point $x_j^* \in S^*$, are often available thus raising the question of whether S^* can be estimated from $\tilde{X}$. An important example of this situation is record linkage [12, 19, 3], where the objective is to integrate data from independent sources, $\tilde{x}_1 \in \mathbb{R}^m, \ldots, \tilde{x}_n \in \mathbb{R}^m$, for subsequent data analysis. Since the entries of different records $\tilde{x}_i$'s are collected separately, the data matrix $\tilde{X}$ is *unlabeled* in the sense that, the entries of its ith row do not necessarily correspond to the same entity. Such kind of unlabeled data $\tilde{X}$ also arise in the context of data privatization, where the data provider anonymizes the original data X^* by permuting

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

each column of X^* prior to release [10, 15]. Data re-identification is a concern since companies with privacy policies, health care providers, and financial institutions may release the collected data after anomymization. Understanding the fundamental limits of re-identifying the original data X^* from the released ones $\tilde{X}$ is essential for striking a balance between data privacy and data preservation, the subject of a plenary talk in the 2019 International Conference on Machine Learning [2].

Contributions. In this paper we consider the recovery of X^* from its unlabeled version $\tilde{X}$, which we term *Unlabeled Principal Component Analysis* (UPCA). We make the following contributions: 1) We establish that as long as $r = \text{rank}(X^*) < \min\{m, n\}$ and X^* is sufficiently generic, then up to a permutation of its rows X^* is the only matrix of rank less than or equal to r that is compatible with $\tilde{X}$. This asserts that UPCA is a well-posed problem, since the inherent ambiguity of whether $\tilde{X}$ comes from X^* or a row-permuted version of X^* is in most cases practically harmless. 2) We establish that in this basic formulation UPCA is a purely algebraic problem, by exhibiting a polynomial system of equations parametrized by $\tilde{X}$, whose solutions are all the row-permutations of X^*; solving the UPCA problem amounts to obtaining one such solution. 3) Inasmuch as solving this polynomial system is in principle NP-hard, we introduce an efficient algorithmic pipeline for the practically relevant case where a significant part of the data have undergone the same permutation, while the rest of the points have been permuted arbitrarily; in the case of record linkage this would correspond to one of the records having much larger size than the others. The first stage of the pipeline employs PCA methods with robustness to outliers [40, 31, 23, 41, 34, 44, 17] to produce an estimate $\hat{S}$ of S^* from $\tilde{X}$; the second stage of the pipeline uses unlabeled sensing methods [28, 30, 33, 22] to furnish an estimate $\hat{X}$ of X^* from $\hat{S}$ and $\tilde{X}$. 4) We introduce a simple but efficient algorithm for unlabeled sensing based on least-squares with recursive filtrations (Algorithm 2). 5) We assess our algorithmic pipeline on synthetic data, face images, educational and medical records with encouraging results.

Related work. There is a large literature on application-specific problems that involve lack of correspondences, e.g. in computer vision or statistics; here we just review four recent methods for unlabeled sensing that will be used in this paper. Recall that in unlabeled sensing one is given a subspace $S^* \subset \mathbb{R}^m$ of dimension r and a point $\tilde{x}$ which is some permuted version of a point $x^* \in S^*$ and the goal is to recover x^* from S^* and $\tilde{x}$. A critical distinction among methods in the literature is the sparsity level α of the permutation, that is the ratio of coordinates that are moved by the permutation. The case of *dense* permutations ($\alpha = 1$) is extremely challenging, with existing methods only able to handle small ranks r. We consider two methods known to perform best in this regime. The first one is the algebraic-geometric AIEM method of [33]. This has linear complexity in m and instead concentrates its effort on solving a polynomial system of r equations in r variables to produce an initialization for an expectation maximization algorithm. Currently, this method is efficient for $r \leq 5$ and intractable otherwise. A very different method is CCV-Min of [22], which proceeds via branch-&-bound together with concave minimization and can handle $r \leq 8$, though intractable otherwise. For *sparse* permutations (small α) we review two methods [28, 30]. The ℓ_1-RR algorithm of [28] applies an ℓ_1 robust linear regression relaxation and it works when $\alpha \leq 0.5$. Another approach is the Pseudo-Likelihood method (PL) of [30], which fits a two-component mixture density for each entry of $\tilde{x}$, one accounting for fixed data and the other for permuted data. The fitting is done via a combination of hypothesis testing, reweighed least-squares, and alternating minimization and the it tolerates up to $\alpha = 0.7$ is sensitive to the particular basis of S^* used to generate $\tilde{x}$.

PCA methods with robustness to outliers will also play a role in this paper and among a large literature we review four modern methods. Recall that for such methods $\tilde{X}$ can be partitioned into inlier points that lie in an unknown r-dimensional subspace $S^* \subset \mathbb{R}^m$ and outlier points that lie away from S^* and the goal is to recover S^* from $\tilde{X}$. The Outlier Pursuit (OP) method of [40] decomposes via convex optimization $\tilde{X}$ into the sum of low-rank and column-sparse parts. The convex method of [41], a successor of [31], referred to as Self-Expr, solves a self-expressive elastic net problem so that each $\tilde{x}_j$ is expressed as an ℓ_2-regularized sparse linear combination of the other points. The self-expressive coefficients define transition probabilities of a random walk on a graph and the average of the t-step transition probability distributions is used as a score for inliers vs outliers, with higher scores expected for the former. The Coherence Pursuit (CoP) method of [23] is based on the following simple but effective principle: with $\tilde{X}_{-j}$ the matrix $\tilde{X}$ with column j removed, for each $\tilde{x}_j$ one computes its coherence $\tilde{X}_{-j}^\top \tilde{x}_j$ with the rest of the points. As it turns out, inliers tend to have coherences of higher ℓ_2-norm than outliers, and the top r $\tilde{x}_j$'s are taken to span $\hat{S}$. Finally, the Dual Principal Component

Pursuit (DPCP) of [34] also known as Fast Median Subspace [17], solves an ℓ_1 non-smooth problem on the Grassmannian via iterated-reweighed least-squares. In contrast to other convex robust-PCA methods, DPCP was shown in [44] to tolerate as many outliers as the *square* of the number of inliers.

2 Theoretical Foundations

Let us denote by $\mathcal{P}_m$ the set of all permutations of coordinates of $\mathbb{R}^m$. We let $X^* = [x_1^* \cdots x_n^*] \in \mathbb{R}^{m\times n}$ be our ground-truth data matrix with rank $r < \min\{m, n\}$ and column-space $S^* = \mathcal{C}(X^*)$, and we suppose that the available data are

$$\tilde{X} = [\tilde{x}_1 \cdots \tilde{x}_n] = [\Pi_1^* x_1^* \cdots \Pi_n^* x_n^*] \in \mathbb{R}^{m\times n}, \tag{1}$$

where each $\Pi_j^* \in \mathcal{P}_m$ is an unknown permutation. Let $\mathcal{P}_m^n = \prod_{i\in[n]} \mathcal{P}_m$ be n ordered copies of $\mathcal{P}_m$, where $[n] = \{1, \ldots, n\}$. For $\underline{\pi} = (\Pi_1, \ldots, \Pi_n) \in \mathcal{P}_m^n$ we set $\underline{\pi}(\tilde{X}) = [\Pi_1\tilde{x}_1 \cdots \Pi_n\tilde{x}_n]$. We pose Unlabeled Principal Component Analysis (UPCA) as the following rank minimization problem:

$$\min_{\underline{\pi}\in\mathcal{P}_m^n} \operatorname{rank} \underline{\pi}(\tilde{X}) \tag{2}$$

First, note that (2) never has a unique solution, because if $\underline{\pi} = (\Pi_1, \ldots, \Pi_n)$ is a solution, then so is $\underline{\pi}' = (\Pi\Pi_1, \ldots, \Pi\Pi_n)$, where $\Pi \in \mathcal{P}_m$ is any permutation. This reveals an inherent ambiguity of UPCA: it is only possible to recover X^* from $\tilde{X}$ up to a permutation ΠX^* of its rows. On the other hand, this is rather harmless in many situations, since ΠX^* is the same dataset as X^* except that the row-features appear now in different order. Thus, our hope in formulating (2) is that the only solutions are of the form $\underline{\pi} = (\Pi\Pi_1^*, \ldots, \Pi\Pi_n^*)$ with Π ranging across $\mathcal{P}_m$ and Π_j^* as in (1). However, without any other assumptions on the data X^*, there could in principle be additional undesired permutations that also give $\operatorname{rank} X^*$, or even worse the minimum rank in (2) could be lower than $r = \operatorname{rank} X^*$. Our results show that for *generic* enough data such pathological situations do not occur and the only solutions to (2) are the ones associated with row-permutations of X^*.

Before stating our results, we make the notion of *generic* precise using some basic algebraic geometry [8, 14]. Let $Z = (z_{ij})$ be an $m \times n$ matrix of variables z_{ij} and $\mathbb{R}[Z] = \mathbb{R}\big[z_{ij} : i \in [m], j \in [n]\big]$ the ring of polynomials in the z_{ij}'s with real coefficients. An algebraic variety of $\mathbb{R}^{m\times n}$ is the set of solutions of a polynomial system of equations in $\mathbb{R}[Z]$. In particular, the set of $(r+1)\times(r+1)$ determinants of Z are polynomials in z_{ij}'s of degree $r+1$ and define the algebraic variety

$$M_r = \{\, X \in \mathbb{R}^{m\times n} | \operatorname{rank} X \leq r \,\},$$

since a matrix $X \in \mathbb{R}^{m\times n}$ has rank at most r if and only if all $(r+1)\times(r+1)$ determinants of X are zero. The algebraic variety $\mathcal{M}_r$ admits a topology, called Zariski topology, which makes it convenient to work with. The closed sets in this topology are the algebraic subvarieties of $\mathcal{M}_r$, that is matrices of rank $\leq r$ which in addition satisfy certain other polynomial equations in $\mathbb{R}[Z]$. For example, the set of matrices of rank at most $r-1$ is a proper closed subset of $\mathcal{M}_r$, because in addition to the equations defining $\mathcal{M}_r$, it is further defined requiring all $r \times r$ determinants to be zero. Open sets in $\mathcal{M}_r$ are defined as complements of closed sets or equivalently they are defined by requiring that certain sets of polynomial equations are not all simultaneously zero. For example, the set of matrices of rank exactly equal to r is a proper open subset of $\mathcal{M}_r$ defined by the non-simultaneous vanishing of all $r \times r$ determinants of Z; a matrix has rank r if and only if all $(r+1)\times(r+1)$ determinants are zero and least one $r \times r$ determinant is non-zero. Now, the algebraic variety $\mathcal{M}_r$ is irreducible in the sense that it can not be described as the union of two proper algebraic subvarieties of it. A consequence of this is that non-empty open sets of $\mathcal{M}_r$ have the very important property of being topologically dense. This means that given a non-empty open set $\mathcal{U} \subset \mathcal{M}_r$ and a point $X \in \mathcal{M}_r$, every neighborhood of X intersects $\mathcal{U}$. It follows that under any non-degenerate continuous probability measure on $\mathcal{M}_r$ a non-empty Zariski-open set of $\mathcal{M}_r$ has measure 1. As an example, the set of matrices in $\mathcal{M}_r$ of rank r is non-empty and open and thus dense. If there exists a dense open set of matrices in $\mathcal{M}_r$ that satisfy a certain property, then we say that a *generic* matrix in $\mathcal{M}_r$ satisfies this property. For example, a generic matrix in $\mathcal{M}_r$ has rank r. Hence a randomly sampled matrix in $\mathcal{M}_r$ under a continuous probability measure will have rank r with probability 1.

Our main theoretical result is:

Theorem 1. *For X^* a generic matrix in $\mathcal{M}_r$, we have that* $\operatorname{rank} \underline{\pi}(\tilde{X}) \geq r$ *for any $\underline{\pi} \in \mathcal{P}_m^n$, with equality if and only if $\underline{\pi}(\tilde{X}) = \Pi X^*$ for some $\Pi \in \mathcal{P}_m$.*

Theorem 1 says that for $X^* \in \mathcal{M}_r$ generic, and up to a permutation of the coordinates of $\mathbb{R}^m$, S^* is the unique r-dimensional subspace that explains the data $\tilde{X}$ in the UPCA sense, and $r = \operatorname{rank} X^*$ is the minimum objective in (2).

How one can go about solving the discrete optimization problem (2)? In general, brute force selection of the Π_j's has complexity $O\big((m!)^n\big)$, which is out of the question. On the other hand, problem (2) has a rich algebraic structure, which we harvest by showing that X^*, up to a permutation of its rows, arises as the unique solution to a polynomial system of equations. To begin with, for each $j \in [n]$ and each $\ell \in [m]$ we define the following column-symmetric polynomials of $\mathbb{R}[Z]$:

$$\bar{p}_{\ell,j}(Z) = \sum_{i \in [m]} z_{ij}^{\ell}$$

$$p_{\ell,j}(Z) = \bar{p}_{\ell,j}(Z) - \bar{p}_{\ell,j}(\tilde{X})$$

Note that $\bar{p}_{\ell,j}\big(\underline{\pi}(Z)\big) = \bar{p}_{\ell,j}(Z)$ for any $\underline{\pi} \in \mathcal{P}_m^n$ and thus $\bar{p}_{\ell,j}(\tilde{X}) = \bar{p}_{\ell,j}(X^*)$. Now let us think of $X \in \mathcal{M}_r$ as a product of two matrices of size $m \times r$ and $r \times n$ and let us define another polynomial ring with variables associated to these two factors. For $i = r+1, \ldots, m$, and $k \in [r]$ and $j \in [n]$ we let b_{ik}, c_{kj} be a new set of variables over $\mathbb{R}$. Organize the b_{ik}'s to occupy the $(m-r) \times r$ bottom block of an $m \times r$ matrix B whose top $r \times r$ block is the identity matrix of size r, and the c_{kj}'s into a $k \times n$ matrix $C = (c_{kj})$. For $i \in [m]$ write $b_i^\top$ for the ith row of B and for $j \in [n]$ write c_j for the jth column of C. With $\tilde{x}_{ij}, x_{ij}^*$ the ith coordinate of $\tilde{x}_j, x_j^*$ respectively, we obtain polynomials $q_{\ell,j}$ for $\ell \in [m]$, $j \in [n]$ of $\mathbb{R}[B, C]$ by substituting $z_{ij} \mapsto b_i^\top c_j$ in the $p_{\ell,j}(Z)$'s above:

$$q_{\ell,j}(B, C) = \bar{p}_{\ell,j}(BC) - \bar{p}_{\ell,j}(\tilde{X}) = \sum_{i \in [m]} (b_i^\top c_j)^\ell - \sum_{i \in [m]} \tilde{x}_{ij}^\ell = \sum_{i \in [m]} (b_i^\top c_j)^\ell - \sum_{i \in [m]} {x_{ij}^*}^\ell$$

The set of common roots of all $q_{\ell,j}$'s defines an algebraic variety that depends only on X^*, given by

$$\mathcal{Y}_{X^*} = \big\{(B', C') \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n} \,|\, q_{\ell,j}(B', C') = 0,\ \forall \ell \in [m],\ \forall j \in [n];\ \ B'_{[r],[r]} = I_r\big\},$$

where $B'_{[r],[r]} = I_r$ signifies that the top $r \times r$ block of $B' \in \mathbb{R}^{m \times r}$ is the identity matrix. Let us get a feeling about the points of $\mathcal{Y}_{X^*}$. With $\Pi \in \mathcal{P}_m$, if the column-space $\mathcal{C}(\Pi X^*)$ of ΠX^* does not drop dimension upon projection onto the first r coordinates, then there exists a unique basis B_Π^* of $\mathcal{C}(\Pi X^*)$ with the identity matrix occurring at the top $r \times r$ block. In that case, there is a unique factorization $\Pi X^* = B_\Pi^* C_\Pi^*$ and the point (B_Π^*, C_Π^*) lies in the variety $\mathcal{Y}_{X^*}$ because

$$q_{\ell,j}(B_\Pi^*, C_\Pi^*) = \bar{p}_{\ell,j}(B_\Pi^* C_\Pi^*) - \bar{p}_{\ell,j}(\tilde{X}) = \bar{p}_{\ell,j}(\Pi X^*) - \bar{p}_{\ell,j}(X^*) = \bar{p}_{\ell,j}(X^*) - \bar{p}_{\ell,j}(X^*) = 0.$$

Our second result says that if X^* is generic, then all points of $\mathcal{Y}_{X^*}$ are of this type. That is, they correspond to factorizations $B_\Pi^* C_\Pi^*$ of ΠX^* as Π varies across all permutations:

Theorem 2. *For a generic matrix X^* in $\mathcal{M}_r$ we have*

$$\mathcal{Y}_{X^*} = \big\{(B_\Pi^*, C_\Pi^*) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{r \times n} \,|\, \Pi \in \mathcal{P}_m;\ B_{\Pi,[r],[r]}^* = I_r;\ \Pi X^* = B_\Pi^* C_\Pi^*\big\}$$

Thanks to Theorem 2 we have the following important conceptual finding. Assuming X^* is generic, to obtain X^* up to some permutation of its rows from $\tilde{X}$, one needs to compute an arbitrary root (B', C') of the polynomial system of equations

$$q_{\ell,j}(B, C) = 0,\ \forall \ell \in [m],\ \forall j \in [n] \tag{3}$$

and multiply its factors to get $B'C'$. Developing a polynomial system solver for UPCA would involve two main challenges: attaining robustness to noise and scalability, with the former typically easier to deal with than the latter. We leave such a direction to future research.

We close this section with a special case of interest, where part of the data have undergone the same *dominant* permutation; in fact given the inherent ambiguity of UPCA discussed above, we may as well take this dominant permutation to be the identity matrix I_m of size $m \times m$. To make this precise, we define the multiplicity $\mu(\Pi)$ of a permutation $\Pi \in \mathcal{P}_m$ to be the number of times that Π appears as $\Pi = \Pi_j^*$ in (1) with j ranging in $[n]$. We have:

Theorem 3. *Suppose that* $\mu(I_m) \geq r+1$ *while* $\mu(\Pi) < r$ *for any other* $\Pi \neq I_m$. *Then for a generic* $X^* \in \mathcal{M}_r$, *we have that* S^* *is the unique solution to the following consensus maximization problem*

$$\max_{\dim S \leq r} \ \#\{\tilde{x}_j \,|\, \tilde{x}_j \in S\,;\, j \in [n]\}, \tag{4}$$

where # *denotes size and the maximization is taken over all subspaces* $S \subset \mathbb{R}^m$ *of dimension* $\leq r$.

Theorem 3 says that for sufficiently generic ground truth data X^*, the given data $\tilde{X}$ admit a natural partition into a set of inliers and outliers with respect to the linear subspace S^*:

$$\tilde{X}_{\text{in}} := \{\tilde{x}_j \,|\, \tilde{x}_j \in S^*\}, \quad \tilde{X}_{\text{out}} := \{\tilde{x}_j \,|\, \tilde{x}_j \notin S^*\}$$

Of course we do not know what the partition into inliers and outliers is, because we do not know what S^* is. But the presence of this geometric structure is enough for PCA methods with robustness to outliers to operate on $\tilde{X}$ in order to estimate S^*. The proofs of Theorems 1-3 can be found in the supplementary material, while the rest of the paper proceeds algorithmically building on this insight.

3 Algorithms

Two-stage algorithmic pipeline for UPCA. We saw in the previous section that the UPCA problem (2) is well-defined (Theorem 1) and in principle solvable by a polynomial system of equations (Theorem 2). However, this polynomial system is at the moment intractable to solve even for moderate dimensions. On the other hand, Theorem 3 suggests the following algorithmic pipeline for the case where there is a dominant permutation, which we can take to be the identity as in the theorem. At Stage-I of the pipeline a PCA method with robustness to outliers [40, 31, 23, 41, 34, 44, 17] is employed to produce an estimate $\hat{S}$ of S^* from $\tilde{X}$. At Stage-II of the pipeline one feeds $\hat{S}$ and $\tilde{X}$ to an unlabeled sensing method [28, 30, 33, 22] which operates point by point returning for every $\tilde{x}_j$ an estimate $\hat{x}_j$ of x_j^*. Here one may choose to threshold the $\tilde{x}_j$'s based on their distance to $\hat{S}$ and apply unlabeled sensing on the outliers only. Alternatively, if extra computational power is available for dispensing with choosing a threshold, one may apply unlabeled sensing on every $\tilde{x}_j$; it is this approach that we follow in the rest of the paper. This procedure is summarized in Algorithm 1.

Algorithm 1 Two-stage Algorithmic Pipeline for UPCA

Input: observed data matrix $\tilde{X}$, rank r
estimate $\hat{S}$ of $S^* \leftarrow$ outlier-robust PCA on $\tilde{X}$ ▷ Stage-I
for $j = 1, \ldots, n$ **do** ▷ Stage-II
    estimate $\hat{x}_j$ of $x_j^* \leftarrow$ unlabeled sensing on $(\tilde{x}_j, \hat{S})$
end for
return estimate $\hat{X} = [\hat{x}_1, \ldots, \hat{x}_n]$ of X^*

A new method for unlabeled sensing: Least-Squares with Recursive Filtration (LSRF). Inasmuch as there are very few scalable unlabeled sensing methods, we here propose a simple but comparatively efficient alternative, Algorithm 2. This method is parameter-free and alternates between ordinary least-squares and a dimensionality reduction step that removes the coordinate of the ambient space on which the residual error attains its maximal value, until r coordinates are left.

4 Experimental Evaluation

4.1 Robust-PCA with permutation-induced outliers

We begin by assessing the performance of Stage-I of the pipeline. This entails understanding how different PCA methods with robustness to outliers behave when the outliers are induced by permutations, as in Theorem 3. We consider Self-Expr [41, 31], CoP [23], OP [40], and DPCP [34, 17]; see related work in section 1. We fix $m = 50$, $n = 500$. With $\dim S^* = r = 1 : 1 : 49$, we sample S^* at random from the Grassmannian $\text{Gr}(r, m)$. Then n points x_j^* are sampled at random from the intersection of S^* with the unit sphere of $\mathbb{R}^m$ to yield X^*. Let n_{in} be the number of

Algorithm 2 Unlabeled Sensing via Least-Squares with Recursive Filtration (LSRF)

Input: permuted point $\tilde{x}_j$, basis B^* of subspace S^*
$v^{(0)} \leftarrow \tilde{x}_j, A^{(0)} \leftarrow B^*$
for $k = 1, \ldots, m - r$ **do**
  $c \leftarrow A^{(k-1)\dagger} v^{(k-1)}$
  $i' \leftarrow \text{argmax}_i \, |v_i^{(k-1)} - A_i^{(k-1)} c|$
  remove the i'th entry of $v^{(k-1)}$ to get $v^{(k)}$
  remove the i'th row of $A^{(k-1)}$ to get $A^{(k)}$
end for
return estimate $\hat{x}_j = A^{(m-r)} A^{(m-r)\dagger} v^{(m-r)}$ for x_j^*

inliers $\tilde{X}_{in}$ and n_{out} the number of outliers $\tilde{X}_{out}$, with $n_{in} + n_{out} = n$. We consider outlier ratios $n_{out}/n = 0.1 : 0.1 : 0.9$. With fixed n_{out}/n, we set Π_j^* to the identity for $j \in [n_{in}]$ and set Π_j^*'s for $j > n_{in}$ as follows. First, we consider both dense ($\alpha = 1$) and sparse ($\alpha = 0.1$) permutations. For a fixed α, we obtain Π_j^* by randomly choosing αm coordinates for each x_j^* and applying a random permutation on those coordinates. As evaluation metric we use the largest among all r principal angles between $\hat{S}$ and S^*, denoted by $\theta_{max}(S^*, \hat{S})$. Recall $\theta_{max}(S^*, \hat{S}) = 0$ if and only if $\hat{S} = S^*$.

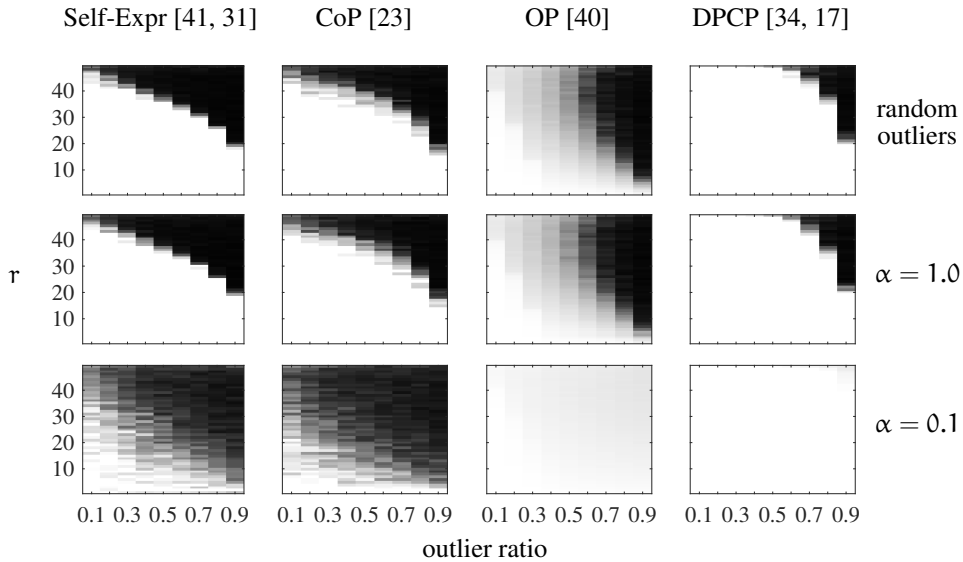


Figure 1: $\theta_{max}(S^*, \hat{S})$ in Stage-I: outlier ratio vs. rank phase transitions for various PCA methods with robustness to outliers.

Figure 1 depicts the *outlier-ratio versus rank* phase transitions, where to calibrate the analysis with what we know about these methods from prior work, we have included in the top row of the figure the phase transitions for outliers randomly chosen from the unit sphere. By reading that top row we recall: i) DPCP has overall the best performance across all ranks and all outlier ratios, ii) OP identifies correctly S^* only in the low rank low outlier-ratio regime, as expected from its conceptual formulation, and iii) CoP and Self-Expr, even though low-rank methods in spirit, they have accuracy similar to each other and considerably better than OP. We also note that OP is the fastest method requiring $0.51sec$ for the computation of a single phase transition plot, Self-Expr is the slowest with $752sec$ and DPCP and OP take $1.31sec$ and $5.62sec$, respectively[1]. Now let us look at what happens for permutation-induced outliers. For $\alpha = 1$, where the permutations move all the coordinates of the points they are corrupting, we see that the phase transition plots are practically the same as for random outliers. In other words, obtaining the outliers by randomly permuting all coordinates of inlier points, with different permutations for different outliers, seems to be yielding an outlier set as generic for the task of subspace learning as sampling the outliers randomly from the unit sphere. A

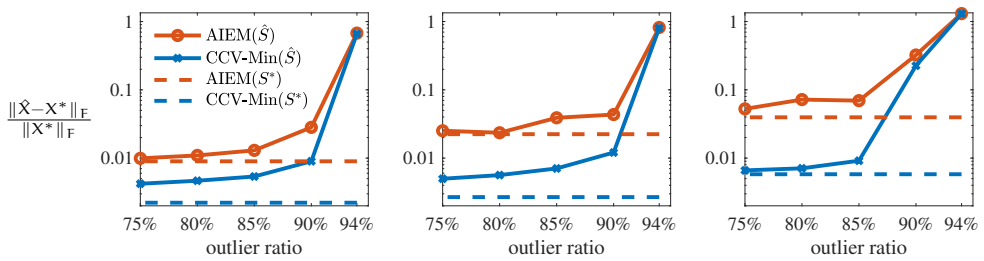
[1]Experiments are run on an Intel(R) i7-8700K, 3.7 GHz, 16GB machine.

second interesting phenomenon is observed when the permutation ratio is decreased to $\alpha = 0.1$. In that regime the methods exhibit two very different trends. On one hand, CoP and Self-Expr appear to break down, which is expected, because as the permutations become more sparse, the outlier points become more coherent with the rest of the data set. On the other hand, the accuracy of DPCP and OP improves for sparser permutations; a justification for this is that both methods get initialized via the SVD of $\tilde{X}$, which yields a subspace closer to S^* for smaller α. An interesting research direction is to analyze the theoretical guarantees of these methods for this specific type of outliers.

4.2 UPCA on synthetic data

Next, we evaluate the UPCA pipeline of Algorithm 1 on synthetic data. We keep $m = 50$ as before, and add spherical noise corresponding to a fixed SNR of 40dB. We get the estimate $\hat{S}$ of S^* via DPCP [34, 17] in Stage-I and apply the unlabeled sensing methods [33, 22, 28, 30] and Algorithm 2 in Stage-II to get $\hat{X}$ from $\hat{S}$ and $\tilde{X}$. We distinguish between dense and sparse permutations.

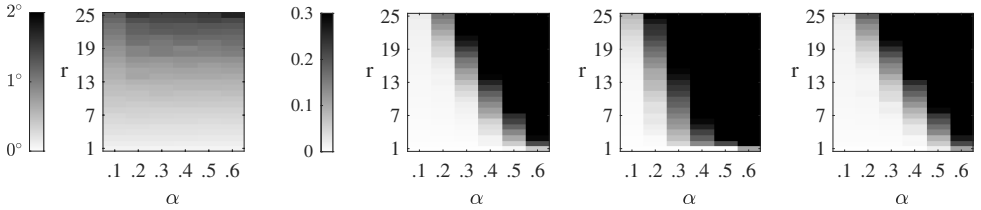
UPCA with dense permutations ($\alpha = 1$). Figure 2 depicts the relative estimation error of $\hat{X}$ for different outlier ratios from 75% (25 inliers) to 94% (6 inliers) and ranks $r = 3, 4, 5$. To assess the overall effect of the quality of $\hat{S}$, we use two versions of AIEM and CCV-Min. The first, denoted by AIEM($\hat{S}$) and CCV-Min($\hat{S}$), uses as input the estimated subspace $\hat{S}$, while the second version, AIEM(S^*) and CCV-Min(S^*), uses the ground-truth subspace S^*. Note that the estimation error of AIEM(S^*)/CCV-Min(S^*) is independent of the outlier ratio. On the other hand, the estimation error of AIEM($\hat{S}$)/CCV-Min($\hat{S}$) depends on the outlier ratio through the computation of $\hat{S}$. Indeed, $\hat{S}$ is expected to be closer to S^* for smaller outlier ratios, as we already know from Figure 1. In particular, for up to 75% outliers the estimation error of AIEM($\hat{S}$)/ CCV-Min($\hat{S}$) coincides with that of AIEM(S^*)/ CCV-Min(S^*), indicating an accurate estimation of S^*. At the other extreme, for 94% outliers both AIEM($\hat{S}$)/ CCV-Min($\hat{S}$) break down, indicating that the estimation of $\hat{S}$ failed. Finally, note that CCV-Min has at least half order of magnitude smaller estimation error than AIEM. This is due to our specific choice of the branch & bound CCV-Min parameters which control the trade-off between accuracy and running time; for example, for $r = 3$ and 75% outliers, AIEM runs in $42msec$ with 1% error, while CCV-Min needs about $15sec$ to bound $\hat{X}$ 0.42% away from X^*.



(a) $r = 3$ (b) $r = 4$ (c) $r = 5$

Figure 2: UPCA (Algorithm 1) for dense permutations ($\alpha = 1$) with $\hat{S}$ produced by DPCP [34, 17] at Stage-I and $\hat{X}$ produced by AIEM [33] or CCV-Min [22] at Stage-II.

UPCA with sparse permutations ($\alpha \leq 0.6$). Figures 3b-3d show the relative estimation error of UPCA for $\alpha = 0.1 : 0.1 : 0.6$, rank $r = 1 : 1 : 25$ and outlier ratio fixed to 90%, with $\hat{S}$ computed in Stage-I by DPCP [34, 17] and $\hat{X}$ computed via ℓ_1-RR [28], PL [30] or Algorithm 2 from $\tilde{X}$ and $\hat{S}$ in Stage-II. It is important to note that $r = 25 = m/2$ is the largest rank for which unique recovery of X^* is theoretically possible [35, 36, 32, 9]. Figure 3a shows that $\theta_{\max}(S^*, \hat{S})$ always stays below 2°, indicating the success of DPCP. Now ℓ_1-RR and Algorithm 2 have similar accuracy, but Algorithm 2 is more efficient than ℓ_1-RR, considering that computing $\hat{X}$ takes $0.3sec$ seconds for Algorithm 2 and $1.5min$ for ℓ_1-RR. Even though PL delivers $\hat{X}$ in $1sec$, it is not performing as well, which we attribute to its sensitivity on the particular basis of S^* that is used to generate the data; this is not available here since DPCP returns the specific basis of dual principal components.



(a) $\theta_{\max}(S^*, \hat{S})$ (b) ℓ_1-RR [28] (c) PL [30] (d) Algorithm 2

Figure 3: Estimation error $\frac{\|X^* - \hat{X}\|_F}{\|X^*\|_F}$ of UPCA (Algorithm 1) for sparse permutations ($\alpha \leq 0.6$) and outlier ratio 90%, with $\hat{S}$ computed by DPCP [34, 17] in Stage-I and $\hat{X}$ computed by ℓ_1-RR [28], PL [30] or Algorithm 2 in Stage-II.

4.3 UPCA on face images

In this section we offer a flavor of how the ideas discussed so far apply in a high-dimensional example with real data. We use the well-known database Extended Yale B [13], which contains fixed-pose face images of distinct individuals, with 64 images per individual under different illumination conditions. It is well-established that the images of each individual approximately span a low-dimensional subspace. It turns out that for our purpose the value $r = \dim S^* = 4$ is good enough. Since each image has size 192×168, the images of each individual can be approximately seen as $n = 64$ points x_j^*, $j \in [64]$ of a 4-dimensional linear subspace S^*, embedded in an ambient space of dimension $m = 32256$. In what follows we only deal with the images of a fixed individual. We consider four permutation types corresponding to fully or partially ($\alpha = 0.4$) permuting image patches of size 16×24 or 48×42, as shown in the second column of Figure 4. To generate a fixed number of $n_{out} = 16$ outliers only one out of the four permutation types is used for each trial. The original images (inliers) together with the ones that have undergone patch-permutation (outliers) are given without any inlier/outlier labels, and the task is to restore all corrupted images. This is a special case of visual permutation learning, recently considered using deep networks [24, 25].

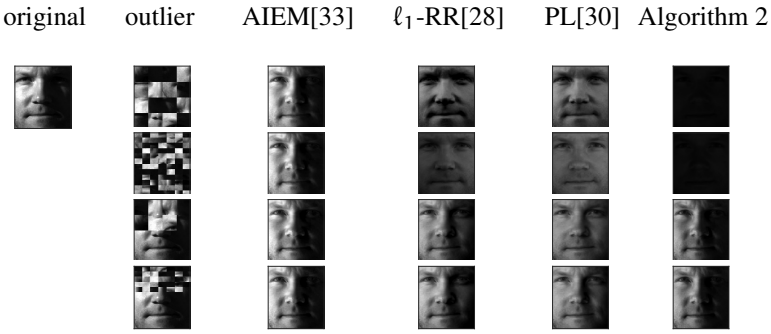


Figure 4: UPCA on the face dataset Extended Yale B.

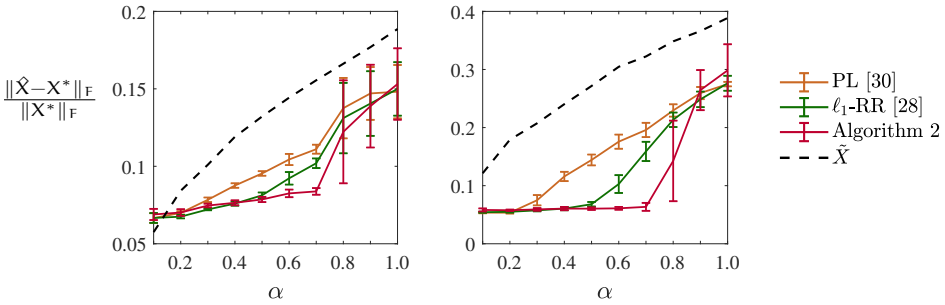
The first column of Figure 4 shows an original image, and the second column shows the corresponding outlier obtained by applying a sample permutation for each of the four different permutation types. Columns three to six give the corresponding point in the output of UPCA for different unlabeled sensing methods and $\hat{S}$ computed by DPCP [34, 17]; CCV-Min [22] is not included as branch-&-bound becomes prohibitively expensive for such large m. All methods return images very close to the original one, suggesting the accuracy of the DPCP estimate $\hat{S}$. Notably, AIEM [33] rather satisfactorily restores the original image regardless of permutation type. Algorithm 2 performs better than ℓ_1-RR when the permutation is sparse (rows $3, 4$), but their performance slightly degrades as expected when the permutation is dense (rows $1, 2$). Finally, PL [30] uniformly fails to capture the illumination of the original image, a fact that we attribute to its aforementioned sensitivity. Overall though, the performance of these methods is surprisingly good, especially if one takes into consideration that they are very efficient, requiring only 0.2sec (AIEM), 7sec (ℓ_1-RR), 0.2sec (PL)

and $10sec$ (Algorithm 2) seconds, discounting the DPCP step, which costs $0.1sec$, regardless of permutation type. We find this to be particularly encouraging, given that existing deep network architectures for visual permutation learning such as [25] are based on branch-&-bound and thus have in principle an exponential complexity in the number of permuted patches.

4.4 UPCA on data re-identification

Finally, we evaluate the UPCA Algorithm 1 for the task of re-identification (Section 1) using real educational and medical records and simulated permutations for various sparsity levels α, thus emulating a privacy protection scenario. Both of the datasets that we use contain no personally identifiable information. DPCP [34, 17] computes $\hat{S}$ in Stage-I and ℓ_1-RR [28], PL [30] or Algorithm 2 produce $\hat{X}$ in Stage-II.

The first dataset consists of the test scores of $m = 707$ high-school students on 6 subjects during two different periods, together with the sum of the score tests for each period, thus $n = 14$. For 7 out of 14 tests we apply random permutations of the student indices and thus have 50% outliers. With $r = 3$, the relative estimation errors on the score records are shown in Figure 5a. The black dashed line depicts the relative difference between the observed data $\tilde{X}$ and the original data X^*, which as expected increases for higher α's. The performance of ℓ_1-RR, PL and Algorithm 2 is in alignment with our earlier findings in that Algorithm 2 tends to have a superior performance and PL is the least competitive. All these methods apply in principle for sparse permutations and thus their accuracy naturally degrades for large α.



(a) high-school scores (b) breast tumor features

Figure 5: Relative estimation error on real data in de-anonymization.

The second dataset consists of all the benign cases in Breast Cancer Wisconsin (Diagnostic) [11]. It has $m = 357$ patients and $n = 30$ features of a breast mass digitized image for each patient. We randomly permute the patient indices for 15 of the features thus having 50% outliers and set $r = 4$. Figure 5b shows the relative estimation error of $\hat{X}$ for various permutation sparsity levels α, with the unlabeled sensing methods exhibiting the same trend as before. Remarkably, for $\alpha = 0.7$, the UPCA Algorithm 1 incorporating Algorithm 2 in Stage-II reduces the original error of the data $\tilde{X}$ from 32.24% to 6.35% in $0.5sec$, as opposed to 15.90% and 19.57% when ℓ_1-RR [28] or PL [30] are incorporated, respectively.

5 Discussion

Some interesting conclusions can be drawn from this work regarding privacy protection. First, partial permutations appear to be not secure, since, as we have seen, permuting only a subset of the features enables re-identification. Re-identification is also possible for the same reason if a database is fully permuted but another uncorrupted database with the same subjects is accessible or present in an assembled dataset. On the other hand, fully permuted data are NP-hard to re-identify and thus suggest a secure means for attaining privacy. A related interesting direction for future research is the case where on top of permutations one has missing entries, a problem that we term unlabeled matrix completion. The missing entries, may either arise naturally, since for example not all same subjects are common across databases, but they could also serve as part of a privacy protection protocol.

References

[1] A. Abid and J. Zou. A stochastic expectation-maximization approach to shuffled linear regression. In *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 470–477, 2018.

[2] J. M. Abowd. Stepping-up: The census bureau tries to be a good data steward in the 21stcentury. 2019.

[3] M. Antoni and R. Schnell. The past, present and future of the german record linkage center. *Jahrbücher für Nationalökonomie und Statistik*, 239(2):319 – 331, 2019.

[4] D. Bertsimas and M. L. Li. Fast exact matrix completion: A unified optimization framework for matrix completion. *Journal of Machine Learning Research*, 21(231):1–43, 2020.

[5] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[6] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[7] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[8] D. Cox, J. Little, and D. OShea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer Science & Business Media, 2013.

[9] I. Dokmanic. Permutations unlabeled beyond sampling unknown. *IEEE Signal Processing Letters*, 26(6):823–827, 2019.

[10] J. Domingo-Ferrer and K. Muralidhar. New directions in anonymization: Permutation paradigm, verifiability by subjects and intruders, transparency to users. *Information Sciences*, 337-338:11 – 24, 2016.

[11] D. Dua and C. Graff. UCI machine learning repository, 2017.

[12] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[13] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[14] J. Harris. *Algebraic Geometry: A First Course*, volume 133. Springer Science & Business Media, 1992.

[15] X. He, Y. Xiao, Y. Li, Q. Wang, W. Wang, and B. Shi. Permutation anonymization: Improving anatomy for privacy preservation in data publication. In *PAKDD Workshops*, 2011.

[16] D. Hsu, K. Shi, and X. Sun. Linear regression without correspondence. In *Advances in Neural Information Processing Systems*, 2017.

[17] G. Lerman and T. Maunu. Fast, robust and non-convex subspace recovery. *Information and Inference: A Journal of the IMA*, 7(2):277–336, 2018.

[18] S. Marano and P. Willett. Making decisions by unlabeled bits. *IEEE Transactions on Signal Processing*, 68:2935–2947, 2020.

[19] K. Muralidhar. Record re-identification of swapped numerical microdata. *Journal of Information Privacy and Security*, 13(1):34–45, 2017.

[20] A. Nejatbakhsh and E. Varol. Neuron matching in c. elegans with robust approximate linear regression without correspondence. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2837–2846.

[21] A. Pananjady, M. J. Wainwright, and T. A. Courtade. Linear regression with shuffled data: Statistical and computational limits of permutation recovery. *IEEE Transactions on Information Theory*, 64(5):3286–3300, May 2018.

[22] L. Peng and M. C. Tsakiris. Linear regression without correspondences via concave minimization. *IEEE Signal Processing Letters*, 27:1580–1584, 2020.

[23] M. Rahmani and G. K. Atia. Coherence pursuit: Fast, simple, and robust principal component analysis. *IEEE Transactions on Signal Processing*, 65(23):6260–6275, 2017.

[24] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould. Deeppermnet: Visual permutation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[25] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould. Visual permutation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):3100–3114, 2019.

[26] A. Schmaltz, A. M. Rush, and S. Shieber. Word ordering without syntax. In *Conference on Empirical Methods in Natural Language Processing*, pages 2319–2324, 2016.

[27] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6830–6841, 2017.

[28] M. Slawski and E. Ben-David. Linear regression with sparsely permuted data. *Electronic Journal of Statistics*, 13(1):1–36, 2019.

[29] M. Slawski, E. Ben-David, and P. Li. Two-stage approach to multivariate linear regression with sparsely mismatched data. *Journal of Machine Learning Research*, 21(204):1–42, 2020.

[30] M. Slawski, G. Diao, and E. Ben-David. A pseudo-likelihood approach to linear regression with partially shuffled data. *Journal of Computational and Graphical Statistics*, 0(0):1–31, 2021.

[31] M. Soltanolkotabi and E. J. Candes. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

[32] M. C. Tsakiris and L. Peng. Homomorphic sensing. In *International Conference on Machine Learning*, 2019.

[33] M. C. Tsakiris, L. Peng, A. Conca, L. Kneip, Y. Shi, and H. Choi. An algebraic-geometric approach for linear regression without correspondences. *IEEE Transactions on Information Theory*, 66(8):5130–5144, 2020.

[34] M. C. Tsakiris and R. Vidal. Dual principal component pursuit. *The Journal of Machine Learning Research*, 19(1):684–732, 2018.

[35] J. Unnikrishnan, S. Haghighatshoar, and M. Vetterli. Unlabeled sensing: Solving a linear system with unordered measurements. In *Annual Allerton Conference on Communication, Control, and Computing*, pages 786–793, 2015.

[36] J. Unnikrishnan, S. Haghighatshoar, and M. Vetterli. Unlabeled sensing with random linear measurements. *IEEE Transactions on Information Theory*, 64(5):3237–3253, May 2018.

[37] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy. Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery. *IEEE Signal Processing Magazine*, 35(4):32–55, 2018.

[38] G. Wang, S. Marano, J. Zhu, and Z. Xu. Target localization by unlabeled range measurements. *IEEE Transactions on Signal Processing*, 68:6607–6620, 2020.

[39] Y. Xie, Y. Mao, S. Zuo, H. Xu, X. Ye, T. Zhao, and H. Zha. A hypergradient approach to robust regression without correspondence. In *International Conference on Learning Representations*, 2021.

[40] H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, 2012.

[41] C. You, D. P. Robinson, and R. Vidal. Provable self-representation based outlier detection in a union of subspaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3395–3404, 2017.

[42] H. Zhang and P. Li. Optimal estimator for unlabeled linear regression. In *37th International Conference on Machine Learning*, pages 11153–11162, 2020.

[43] T. Zhang and Y. Yang. Robust pca by manifold optimization. *Journal of Machine Learning Research*, 19(1):3101–3139, 2018.

[44] Z. Zhu, Y. Wang, D. P. Robinson, D. Naiman, R. Vidal, and M. C. Tsakiris. Dual principal component pursuit: Improved analysis and efficient algorithms. In *Advances in Neural Information Processing Systems*, pages 2171–2181, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] As discussed in the introduction and the discussion sections, our work has positive impact on the society because it concerns data privacy and protection.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] We include the proofs in supplemental material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code, data, and instructions are included in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specify the parameter setting in the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 5.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We specify the running machine in a footnote in the paper.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include the code in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The datasets are public and anonymized.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The data set we use is anonymized, without any personally identifiable information.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]