Reinforced Robust Principal Component Pursuit

Pratik Prabhanjan Brahma, Yiyuan She, Shijie Li, Jiade Li, and Dapeng Wu, Fellow, IEEE

Abstract—High-dimensional data present in the real world is often corrupted by noise and gross outliers. Principal component analysis (PCA) fails to learn the true low-dimensional subspace in such cases. This is the reason why robust versions of PCA, which put a penalty on arbitrarily large outlying entries, are preferred to perform dimension reduction. In this paper, we argue that it is necessary to study the presence of outliers not only in the observed data matrix but also in the orthogonal complement subspace of the authentic principal subspace. In fact, the latter can severely skew the estimation of the principal components. A reinforced robustification of principal component pursuit is designed in order to cater to the problem of finding out both types of outliers and eliminate their influence on the final subspace estimation. Simulation results under different design situations clearly show the superiority of our proposed method as compared with other popular implementations of robust PCA. This paper also showcases possible applications of our method in critically tough scenarios of face recognition and video background subtraction. Along with approximating a usable low-dimensional subspace from real-world data sets, the technique can capture semantically meaningful outliers.

Index Terms—Manifold optimization, nonconvex penalties, robust principal component analysis (PCA), subspace learning.

I. INTRODUCTION

There has been a huge amount of research done with the objective of recovering low-dimensional structures from high-dimensional data, such as images, videos, text documents, and bioinformatics. Due to the curse of dimensionality and algorithm scaling issues, it is important to estimate a low-dimensional representation of the data which can ease the subsequent learning tasks. Classical principal component analysis (PCA) [1] tries to find out the directions of maximum variation and is undoubtedly the most popular method to serve the purpose. PCA can also be framed as a low-rank approximation problem [2]. However, due to the $\ell_2$ loss used in the objective function of PCA, it becomes impractical and ineffective when the observations are grossly corrupted with outliers and noise.

To robustify PCA, [2] proposed a new model assuming that the design matrix $X \in \mathbb{R}^{n \times p}$ of $p$-dimensional variables can be written in the form of $X = B + O$, where $B$ is the low-rank matrix approximation with $\text{rank}(B) \leq r$ and $O$ is the sparse outlier matrix with few large-valued entries in random positions. The joint recovery of $B$ and $O$ can be characterized by minimizing the objective function $\text{rank}(B) + \lambda \gamma_0(O)$, where $\| \cdot \|_0$ denotes the elementwise $\ell_0$ norm which is the number of all nonzero entries in the matrix. In order to facilitate the computation and analysis, [2] provides a convex relaxation of the formulation called principal component pursuit (PCP) as

$$\min_{B, O} \|B\|_* + \lambda \gamma_0(O) \text{ s.t. } X = B + O$$  \hspace{1cm} (1)

where $\| \cdot \|_*$ denotes the matrix nuclear norm (sum of all singular values), and $\| \cdot \|_1$ denotes the elementwise $\ell_1$ norm (sum of the magnitudes of all matrix entries). Following its huge success in robust subspace estimation, there have been various extensions of PCP, such as [3]–[5]. An exhaustive survey of robust PCA methods is provided in [7]. Such models have found widespread applications, particularly in image and video analysis like learning the invariant subspace in face recognition [6] and subtracting fixed low-rank background from video to do object tracking [8], [9].

However, PCP and other techniques built on top of it face a lot of limitations. First, the $\ell_1$ penalty applied on the sparse outlier matrix is only the best convex approximation to the sparsity-enforcing $\ell_0$ norm. Though popularly used in lasso type of problems, it cannot handle collinearity and may end up in inconsistent selection according to the irrepresentable conditions problem [10]. It also introduces unnecessarily extra bias in the final estimation. This has been the motivation behind switching to nonconvex penalties, such as SCAD [11] and $\ell_0 + \ell_2$ [12]. Second, it has been noticed that the tuning of the parameter $\lambda$ is difficult and requires experimental expertise, since it does not speak directly of the number of outliers in the given data. Instead of using a Lagrange multiplier, it is perhaps more suitable to apply direct threshold in form of constraining inequalities on the number of outlying elements as that makes more sense from a practitioner’s perspective. This also corresponds to the popularly used variable screening techniques [14] where only a set of features are chosen based on selection rules.

But the inability of PCP to deal with certain types of outliers is more serious than the above two issues. Although PCP can effectively identify outliers in the original observation space (OS) as an additive component matrix, it may fail in the presence of potential outliers in the orthogonal complement (OC) subspace, which is the space orthogonal to the authentic PC subspace. This paper shows that any algorithm with the aim of robust subspace learning should specifically identify and remove the outliers in OC subspace, because these...
can skew the subspace estimation drastically. While PCP and its variants treat all of the outliers from the perspective of the OS, there may exist some OC outliers which go unnoticed as good observations.

Section II raises this concern and explains the various types of situations with outliers and their effects using both simulation and real-world data. Section III attempts to come up with a mathematical model to achieve the estimation of principal subspace in presence of all kinds of outliers. A set of algorithms are developed in Section IV to solve a truly robust version of PCA, which is named the reinforced robust PCP (R^2-PCP). The computational algorithms are then used in Section V on simulation data to show the performance advantages of R^2-PCP against various other methods for robust PCA. The algorithms are also tested on machine learning tasks such as face recognition and video background subtraction to show improvement over classical PCA and robust PCP. Finally, we conclude this paper by summarizing the various results and remarks observed from the different experiments.

II. WHAT ARE OUTLIERS

Outliers are arbitrarily large-valued measurements, which do not characterize the true data samples. While the unobservable noise is assumed to have a fast-decaying Gaussian probability distribution, outliers are observations or values that are considerably different from the majority of the data and usually follow heavy-tailed distributions. If the observed examples are stacked as rows in data matrix as shown in Fig. 1, either some of the particular entries may be affected due to additive outliers or entire rows (whole examples) are corrupted. On the basis of location of the outliers in the data matrix, outliers can be classified into two types.

1) r-Type Outliers: Those observations or examples which are wholly corrupted and do not belong to the true principal data subspace are termed rowwise (r-type) outliers.

2) e-Type Outliers: When some of the entries, but not all, at arbitrary locations in the design matrix have outlying values, they are considered to be elementwise (e-type) outliers.

As will be shown later, PCP performs poorly when there are too many r-type outliers in the data. Also, the outliers may not be always visible in the observed design matrix. A deeper analysis of the different types of outliers is shown in Fig. 2. The true data in the observable R^3 space is actually a finite linear R^1 manifold or just a straight line. This direction, along which the data varies the maximum, is also the PC subspace. Our objective is to accurately identify this subspace in presence of noise and outliers.

Point O is the data center and, for esthetic reasons, it is taken to be the origin itself. If an outlier lies along the PC, it does not adversely affect the PC subspace estimation. The subspace estimation is largely skewed when there are outliers (or outlying components) away from the PC subspace. The directions orthogonal to the PC subspace span the OC subspace. In Fig. 2, it is the R^2 plane to which the true data line acts as a normal. Data points like A, with large enough OC projection components, are the ones that cause maximum damage to the performance of any robust PCA technique. A challenge is that in the OS, their coordinates may lie within the range of variation in actual data. Therefore, these outliers may appear as good observations. The big question is whether PCP and its variants can remove the influence of such outliers from the estimation of the true subspace. Our simulation results in the later sections disclose the weakness of these algorithms in handling such cases. We also go ahead to argue that such outliers can practically exist in real-world data and it is appropriate to deal separately with these OC outliers after transforming the OS properly.

A taxonomy that can ease the handling of outliers in our subsequent mathematical modeling is considered here. While the previous classification was on the basis of location in the matrix, the following classification considers the subspace in which these are most prominent. The types of outliers in a robust PCA setup can be classified into two types.

1) OS Outliers: Those entries in the data matrix which have relatively anomalous values as compared to the range of variation in the true data are considered to be OS outliers.

2) OC Outliers: The observations which have outlying magnitude when projected onto the OC subspace and
which have also not been treated as visible outliers in the OS are considered to be OC outliers.

In our subsequent analysis, OC outliers are usually taken to be the whole observation itself and hence are \( r \)-type in nature. Data point A in Fig. 2 is a good example of an OC outlier. It is interesting to note that if we project all the data points onto the OC subspace, the true data samples capture a small volume around the data center \( O \). However, the projection of point A has a large row-norm or \( \ell_2 \) Euclidean distance from the origin \( O \). Even if the coordinates of A may lie within the range of variation of the true data samples, its row-norm in OC subspace can still be large enough, especially in high-dimensional scenarios. This makes the whole sample a prominent outlier that is better disregarded from consideration as is common with the robust statistics literature.

Such a branching of outlier types makes it convenient to handle these observations independently. For example, in Fig. 2, point A is a \( r \)-type OC outlier, while point B can be \( e \)-type OS outliers. The PCP type of algorithms assumes just additive outliers \( O \) with large values in the OS, as shown in (1). However, in this paper, along with the OS outlier matrix \( O \), we design a more complete model by including a sparse matrix \( S \) that specifically considers outliers in the OC subspace. Thus, \( r \)-type OC outliers correspond to row sparsity of \( S \). On being projected back to the OS for modeling purposes, we come up with the decomposition \( X = B + O + (\mu^T + S) V^T + E \), where \( \mu \) is the data center vector and \( E \) is additive noise. Section III describes the mathematical model in detail. For all our subsequent analysis, we go with the combination of \( r \)-type OC and \( e \)-type OS outliers because of its semantic implications in real-world situations. However, other combinations can also be easily handled by our design framework.

Computer vision problems involve high-dimensional (according to the number of pixels and channels, such as R, G, B, and depth) image data sets, which usually belong to low-dimensional manifolds. Hence, subspace estimation can help in reducing the dimensionality of the problem and easing subsequent learning tasks. However, these data sets are frequently filled with unnecessary inclusions, which make the estimation difficult. For example, recognition of humans from camera images of their faces may suffer from outliers, such as glasses, beard, or other kinds of occlusions distorting the true facial appearance of the person. In the presence of varying illumination, cast shadows and specular reflections should also be ignored while learning the actual face subspace. In perspective of the data matrix \( X \) formed by putting all the pixel intensity values of a single image in each row, these unwanted occlusions are \( e \)-type outliers as only some of the pixels are affected. With the recent surge of online social networks, recognizing pictures requires a robust training of invariant features defining an individual’s face. Often the picture folders of users, like in Fig. 3, contain face images along with nonface images, such as scenic wallpapers, food, cartoons, and beaches. Also, some of the face pictures, which are heavily jittered or blurred, may hinder the estimation of the true subspace. These images are whole examples which are also \( r \)-type outliers and should be removed while learning the principal subspace.

While PCP would treat all of these outliers as additive aberrations in the OS, a fully robust PCA model should take care of both OS and OC outliers present either as \( r \) or \( e \)-type separately. Reflections, shadows, and occlusions affect certain pixels only and are clearly \( e \)-type outliers in the OS, whereas the nonface images are \( r \)-type OC outliers. The set of hidden factors that describe the images of only the clean faces does not characterize the generation of the nonface images. In other words, the OC outliers are independent of the low-dimensional intrinsic face manifold. Hence, such data points should show prominent components in the space orthogonal to true PC subspace. It can be the case that some of them, like the cartoon in Fig. 3, may seem like genuine face images and thus may have similar features and patterns. However, when the entire data set is projected onto the appropriate OC subspace, the nonface images will lie far away from the small volume occupied by the clean face data. Thus, the advantage of treating these as OC outliers is that they are more clearly distinguishable in the OC subspace as compared with the OS. This hypothesis is later verified in Section V.

Another problem where robust subspace estimation finds a suitable application is tracking moving objects for video surveillance [6]. The static background is usually considered to be a linear low-rank subspace for a fixed camera, while moving objects in the video frames are modeled as outliers. A low-rank matrix approximation of the entire video frame sequence using PCP or any other robust PCA method can estimate the fixed background, subtracting which can help track mobile objects as outliers. Also for nonstationary cameras on mobile devices, trajectory analysis as in [15] can help design a more generalized variant of robust PCA for subtracting the fixed background, which is again a low-rank component in the \( \mathbb{R}^3 \) space, that is with an intrinsic rank less than or equal to 3. Experiments, however, show that most of these methods usually break down in the presence of too many outliers. Also, there might be special situations of blocking the camera view. As can be seen in Fig. 4, videos captured from a camera on top of autonomous cars may contain a large number of moving objects and it might also be the case that the background view...
may get lost for certain time in the video due to obstruction in front of the camera. A robust PCA model should cater to all these possible cases. Fortunately, all of these situations can be suitably modeled under our reinforced $\mathbb{R}^2$-PCP architecture.

### III. Mathematical Modeling

The objective function of PCP in (1) tries to robustify PCA by decomposing the data matrix $X$ into a low-rank component $B$ and a sparse outlier component $O$ in the OS. In [3], the model of robust PCP is extended to $X = B + O + E$ by considering a Gaussian noise term $E$. However, as pointed out in Section II, robust PCA-based approaches may be unable to tackle the OC outliers. While ROC-PCA [31] attempts at modeling the OC outliers, it fails in the presence of both types of outliers. To explicitly characterize the outliers both in the OS and the OC subspace, we describe the complete data model as

$$X = B + O + (1\mu^T + S)V_1^T + E$$

with $\text{rank}(B) \leq r$ and the row space of $B$ is orthogonal to the column space of orthogonal matrix $V_\perp$, i.e., $B = CV_\perp^T$, where $C$ can be an $n \times r$ matrix. In (2), the given $n \times p$ data matrix $X = [x_1, \ldots, x_n]^T$ is decomposed into several components. Let $d := p - r$ be the dimensionality of the OC subspace. Each component can now be explained as follows.

1. $B \in \mathbb{R}^{n \times p}$ is the rank-$r$ ($r \ll p$) component.
2. $O = [o_1, \ldots, o_n]^T = [o_{ij}] \in \mathbb{R}^{n \times p}$ represents the outliers detectable in the OS.
3. $SV_\perp^T$ is the outlier component from the OC subspace projected back onto the OS. Here, $S = [s_1, \ldots, s_n]^T$ is an $n \times d$ outlier matrix describing the outlyingness of each observation in the OC subspace and $V_\perp \in \mathbb{O}^{p \times d}$ (where $\mathbb{O}$ is the Stiefel manifold defined by $V_\perp V_\perp^T = I$) characterizes the OC subspace orthogonal to the rank-$r$ PC subspace. $\mu$ is a $d$-dimensional mean vector for the observations in the OC space.
4. $E$ is the unobservable noise term which is independent entries sampled from a possible Gaussian $\mathcal{N}(0, \sigma^2)$ distribution.

Under the Gaussian assumption of the noise, a maximum likelihood (ML) estimation problem can be formed so as to minimize $\|X - O - B - (1\mu^T + S)V_\perp^T\|_F^2$. First, $V \in \mathbb{O}^{p \times r}$ consists of the top $r$ ideal PC loading vectors of $X$ and $[V, V_\perp] \in \mathbb{O}^{p \times r}$ forms a fully orthonormal basis. Since the row space of $B$ is orthogonal to the column space of $V_\perp$, the low-rank component can also be written as $B = CV_\perp^T$, where $C$ can be an $n \times r$ random matrix. To simplify the mathematical formulation, we consider the decomposition of $X - O$ into the mutually orthogonal PC and OC subspaces as

$$X - O = (X - O)V_\perp V_\perp^T + (X - O)V_\perp V_\perp^T.$$  

With a little further matrix algebra, the objective function can now be restructured as

$$\begin{align*}
\|X - O - B - (1\mu^T + S)V_\perp^T\|_F^2 \\
= \|(X - O)V_\perp - C\|_F^2 + \|(X - O)V_\perp - 1\mu^T - S\|_F^2.
\end{align*}$$

The minimization with respect to $C$ makes the first term in (4) always vanish for any $O$ and $V$ by setting $C = (X - O)V$.

As discussed earlier, the sparsity in $O$ and $S$ should also be considered in the optimization. This motivates the use of general sparsity-enforcing constraints to the loss function. A penalized optimization problem is given as

$$\min_{\mu, O, S, V_\perp} \frac{1}{2} \|(X - O)V_\perp - 1\mu^T - S\|_F^2 + P_O(O; \lambda_O) + P_S(S; \lambda_S)$$

s.t. $V_\perp^TV_\perp = I$

where $P_O(O; \lambda_O)$ and $P_S(S; \lambda_S)$ are general sparsity penalties with $\lambda_O$ and $\lambda_S$ as the regularization parameters, respectively. Let $\ell(V_\perp, \mu, O, S) = (1/2)\|(X - O)V_\perp - 1\mu^T - S\|_F^2$ be the objective function in the (5). Throughout this paper, we refer to the investigation of (5) as $\mathbb{R}^2$-PCP. In addition to offering a robust estimate of the PC subspace $V$ by isolating outliers in the OS as in PCP, $\mathbb{R}^2$-PCP also aims at being resistant to outliers in the transformed OC subspace. Two main sparsity-enforcing ways, elementwise or group form, can be applied to design $P_O$ and $P_S$. All commonly used penalties can be adopted, including $\ell_1$, SCAD [11], elastic net [16] $\ell_0 + \ell_2$ [39], and so on. Since we favor the combination of e-Type $O$ and r-Type $S$ for its wide applicability in real data applications, the penalty on the OS outliers $P_O(O)$ could take the form of

$$\|O\|_0 \leq q_0^O$$

and that for $P_S(S)$ is given by

$$\|S\|_{2,0} \leq q_S^S.$$  

The $\ell_0$ and group $\ell_0$ constraints are used in view of the aforementioned resistance to gross outliers. It is also intuitively easier to tune such threshold parameters instead of dealing with Lagrangian multipliers. A ridge $\ell_2$ penalty can also be added to get smooth and regularized estimates. We will discuss how to design nonconvex optimization algorithms to handle various penalties in Section IV.

### IV. Computational Algorithm

Our approach to solve the ($\mathbb{R}^2$-PCP) problem is to alternatively find the estimates of $(\mu, O, S)$ and $V_\perp$ and use these to compute each other variables iteratively. This is continued for
A fixed number of iterations or until convergence. The overall solver is given in Algorithm 1. Each of the intermediate steps, i.e., $(\mu, O, S)$-opt and $V_\perp$-opt are elaborated in detail in Sections IV-A–IV-C.

Algorithm 1: R²-PCP Overall Solution

**Input:** $X, r$

**Output:** $V_\perp, \mu, O$ and $S$

Initialize $V_\perp^{(0)}, \mu^{(0)}, O^{(0)}$ and $S^{(0)}$.

$i = 0$.  

repeat

\[ i = i + 1 \]

Using $V_\perp^{(i-1)}$,  

do $(\mu, O, S)$-opt.

Using recently estimated $\mu^{(i)}, O^{(i)}$ and $S^{(i)}$, do $V_\perp$-optimization.

until convergence

A. Optimizing $\mu, O,$ and $S$

With fixed $V_\perp$, the optimization reduces to

\[
\min_{\mu, O, S} \frac{1}{2} \| XV_\perp - 1\mu^T - OV_\perp - S \|^2_F + P_O(O) + P_S(S).
\]

(8)

Given $O$ and $S$, the $\mu$-optimization is just an ordinary least square problem, since the penalties are independent of $\mu$. The globally optimal solution can be explicitly calculated as $\mu^o = (1/n)(XV_\perp - OV_\perp - S)^T 1$. However, optimizing $O$ and $S$ with fixed $\mu$ is relatively challenging. Practitioners may favor different forms of sparsity-enforcing penalties for $P_O$ and $P_S$, including elementwise and groupwise variants. In order to provide a general algorithmic framework to solve such optimization problems, we provide an extended version of the thresholding-shrinkage-based $\Theta$-estimators [33].

A thresholding rule, denoted by $\Theta(\cdot; \lambda)$ with $\lambda$ as a parameter, is defined to be an odd monotone unbounded shrinkage function. A rigorous definition can be found in [12]. In general, it is a set of thresholding rules on the variables we are applying our sparsity constraints on. It can cover all local minimum points and have guaranteed statistical performance [33]. The $\Theta$-estimator is used here to solve group penalized problems with possibly nonorthogonal designs such as

\[
\min_{\beta} L(\beta) + \sum_{m=1}^{M} P_m(\| \beta_m \|_2; \lambda_m)
\]

where $L$ is the objective cost function, $\beta_m$ is the $m$th group of the parameter $\beta$ to be estimated, and $P_m$ is penalty function. There is also a universal duality connection between thresholding rules and penalty functions. For example, $P(\beta; \lambda) = \bar{\lambda} \| \beta \|_2^2$ translates into a simple function

\[
\Theta(t; \lambda) = \frac{t}{t + \bar{\lambda}}.
\]

(10)

Similarly, the simple function $\Theta(t; \lambda) = t$ if $||t|| \leq \bar{\lambda}$ and 0 otherwise translates into the hard-thresholding penalty given by $P(\beta; \lambda) = \chi_{||\beta|| < \lambda}(\bar{\lambda} \| \beta \|_2^2/2) + \chi_{||\beta|| \geq \bar{\lambda}} \| \beta \|_2^2/2$.

The analysis in [39] demonstrates that a coordinatewise minimum point of (9) can easily be obtained by performing an iterative multivariate thresholding procedure

\[
\beta_m^{(k+1)} = \hat{\Theta}_m\left(\beta_m^{(k)} - \alpha \frac{\partial L(\beta)}{\partial \beta_m} |_{\beta = \beta^{(k)}} ; \lambda_m \right) (1 \leq m \leq M).
\]

(11)

Such an iterative procedure is guaranteed to converge if we choose $\alpha$ suitably small enough. This strategy covers all thresholding rules, and all practically used penalties (convex or nonconvex). A special example is the group linear model setup, where $y = X\beta + \epsilon$ with Gaussian noise $\epsilon$, and thus, $-L(\beta) = \| y - X\beta \|_2^2$. By setting the step size $\alpha$ to be less than or equal to $1/\| X \|_2^2$, where $\| \cdot \|_2$ denotes the spectral norm, the iterations can guarantee convergence by virtue of estimation theory [36].

When different types of outliers occur in $O$ and $S$, we introduce separate thresholding rules. In the following analysis, we consider the recommended $l_0 + l_2$ as an example. To deal with the $e$-Type $O$ and $r$-Type $S$ combination, we take advantage of the elementwise $\ell_0$ constraint ($\| O \|_0 \leq q_O^e$) and the group constraint ($\| S \|_{2,0} \leq q_S^e$) to replace $P_O$ and $P_S$, respectively. Here, $\| S \|_{2,0}$ refers to the number of nonzero rows in $S$. Thus, (8) with fixed $\mu$ becomes

\[
\min_{O, S} \ell(V_\perp, \mu, O, S) + \eta O_0/2 \| O \|_F^2 + \eta S_2/2 \| S \|_F^2
\]

s.t. $\| O \|_0 \leq q_O^e$ and $\| S \|_{2,0} \leq q_S^e$.

(12)

Note that ridge penalties ($\eta O_0/2 \| O \|_F^2 + \eta S_2/2 \| S \|_F^2$) are added to deal with collinearity and benefit from the bias-variance tradeoff. Typically, the values of $\eta O$ and $\eta S$ can be set small, e.g., $\eta O = \eta S = 1e - 3$. This nonconvex fusion penalty usually gives pretty sparse estimates with good statistical accuracy.

To solve for $O$ with $S$ fixed, it suffices to study

\[
\min_O \ell(V_\perp, \mu, O, S) + \eta O_0/2 \| O \|_F^2 \; \text{s.t.} \; \| O \|_0 \leq q_O^e
\]

(13)

and similarly with $O$ fixed, (12) becomes

\[
\min_S \ell(V_\perp, \mu, O, S) + \eta S_2/2 \| S \|_F^2 \; \text{s.t.} \; \| S \|_{2,0} \leq q_S^e.
\]

(14)

The gradient of $l = (1/2)\| XV_\perp - 1\mu^T - OV_\perp - S \|_F^2$ with respect to $O$ and $S$ is computed as $G_O = -(XV_\perp - 1\mu^T - S - OV_\perp)V_\perp^T$ and $G_S = -(XV_\perp - 1\mu^T - OV_\perp - S)$, respectively. The computational algorithm for solving (13) and (14) relies on two nonlinear quantile thresholds $\Theta^\#$ and $\tilde{\Theta}^\#$, which are defined as follows:

\[
O^{(k+1)} = \Theta^\#(O^{(k)} - \alpha_O G_O^{(k)}; \eta O_0; q_O^e)
\]

(15)

and

\[
S = \tilde{\Theta}^\#((X - O)V_\perp - 1\mu^T; \eta S; q_S^e).
\]

(16)

Usually, $\alpha_O = 1/\| V_\perp \|_2 = 1$ leads to good convergence. Due to the identity design of $S$ in the optimization problem (14), the estimation in (16) gives a globally optimal solution in just one shot. However, $O$ has to be iteratively estimated. We combine the updating of $\mu$, $O$, and $S$ together...
Algorithm 2: \((\mu, O, S)\)-Opt for e-Type \(O\) and r-Type \(S\)

1. \(k \leftarrow 0.\)

2. repeat
   2.1. \(S^{(k+1)} \leftarrow \Theta^\theta((X - O^{(k)})V_\perp - 1\mu^{(k)}T; \eta S; q^0_\perp).\)
   2.2. \(\mu^{(k+1)} \leftarrow \frac{1}{\eta}((X - O^{(k)})V_\perp - S^{(k+1)})T 1.\)
   2.3. \(j \leftarrow 0; \; \hat{O}^{(0)} \leftarrow O^{(i)}.\)

3. repeat
   3.1. \(\hat{O}^{(j+1)} \leftarrow \Theta^\theta(\hat{O}^{(j)}(I - aO)V_\perp V_\perp T) + aO(XV_\perp - 1\mu^{(k+1)})T - S^{(k+1)}V_\perp T; \eta O; q^0_\perp).\)
   3.2. \(j \leftarrow j + 1.\)

4. until \(\|\hat{O}^{(j)} - \hat{O}^{(j-1)}\|_\infty \) is small

5. \(O^{(k)} \leftarrow \hat{O}^{(j)}.\)

6. \(k \leftarrow k + 1.\)

until \(k \) is large enough

and operate alternatively until we obtain decent estimates of all. Algorithm 2 details the entire flow of estimating the concerned variables.

B. Optimizing \(V_\perp\) on a Stiefel Manifold

Given \(\mu, O,\) and \(S,\) the minimization of \(l\) with respect to \(V_\perp\) is equivalent to

\[
\min_{V_\perp \in \mathbb{R}^{p \times d}} \|l(V_\perp)\| = \frac{1}{2}\|(X - O)V_\perp - 1\mu T - S\|^2_2.
\]

(17)

The rank constraint space \(\mathbb{O}^{p \times d} = \{V_\perp \in \mathbb{R}^{p \times d} : V_\perp V_\perp T = I\}\) is geometrically a Stiefel Manifold. A local minima of our objective function in (17) should be reached while restricting the walk over its surface only. One of the various techniques [17] to solve this is given by the ManiOpt package [19], which attempts at a gradient-based iterative estimate of \(V_\perp\).

Here, we adopt a nonmonotone line search scheme together with Barzilai–Borwein (BB) stepsize technique [20]. In comparison with other commonly used inexact line searches, this does not necessarily provide a decent in function value at each step but results in quicker convergence and performs well in large-scale nonlinear optimization. The nonmonotone search scheme performs backtracking only occasionally, for which its computational cost remains lower.

Since \(l\) is smooth in \(V_\perp,\) problem (17) can be viewed as an unconstrained optimization problem. Optimization on the Stiefel manifold \(\mathbb{O}\) requires preserving the orthogonality constraint in updating \(V_\perp\). The updating scheme involves retraction [17], which smoothly maps the tangent space \(T_{V_\perp} \mathbb{O}^{p \times d} = \{\Delta \in \mathbb{R}^{p \times d} : V_\perp^T \Delta + \Delta^T V_\perp = 0\}\) onto the Stiefel manifold \(\mathbb{O}^{p \times d}\) as is done in [17].

Let \(G\) denotes the Euclidean gradient of \(l\) with respect to \(V_\perp,\) i.e., \(G_{ij} = ((\partial l(V_\perp))/\partial V_{\perp ij}).\) For our interest, the Riemannian gradient of \(l\) with respect to \(V_\perp,\) denoted by \(\nabla_{V_\perp}\), is then given by

\[
\nabla_{V_\perp} l = W V_\perp
\]

(18)

with \(W = GV_\perp T - V_\perp G^T\) and \(G = (X - O)^T[(X - O)V_\perp - 1\mu T - S].\)

Let \(V_\perp(\tau)\) be a function determining the new trial point with \(\tau\) as the step size. A valid updating scheme should guarantee that the new trial point lies on the manifold. This is ensured by using a Cayley transformation-based update as in [22]

\[
V_\perp(\tau) = (I + \frac{\tau}{2} W)^{-1}(I - \frac{\tau}{2} W) V_\perp.
\]

(19)

This curve always lies on the manifold \(\mathbb{O}\) for any \(\tau,\) and is also a descent curve passing the point \(V_\perp(0) = V_\perp.\) A proper value for the step size \(\tau\) to guarantee convergence and efficiency in large-scale computation is chosen using BB stepsize update technique. Yet the inversion of the \(p \times p\) matrix \((I + (\tau/2)W)\) in (19) may be computationally expensive when dealing with high-dimensional data. There are many techniques to bypass the matrix inversion but still give the same solution as by (19). When \(d < p/2,\) we write \(W = A_1 A_2^T\) with \(A_1 = [G V_\perp] \) and \(A_2 = [V_\perp - G],\) and apply the matrix inversion formula to get \(V_\perp(\tau) = V_\perp - \tau A_1 (I + \tau A_2^T A_1/2)^{-1} A_2^T V_\perp.\) This update formula involves the inversion of a \(2d \times 2d\) matrix, and turns out to be slightly faster than the original form. In the case of \(d \geq p/2,\) one possible idea is to approximate \(W\) by the product of two low-rank matrices. In the later sections, we also provide batch versions of the algorithm, which help divide its computational burden.

The entire stepwise procedure is outlined in Algorithm 3.

Algorithm 3: \(V_\perp\)-Opt ManiOptPackage

1. \(k \leftarrow 0,\)
2. \(\tau_0 \leftarrow 0.5.\)
3. repeat
   3.1. \(G = (X - O)^T[(X - O)V_\perp - 1\mu T - S].\)
   3.2. \(W = GV_\perp T - V_\perp G^T.\)
   3.3. \(V_\perp = (I + \frac{\tau}{2} W)^{-1}(I - \frac{\tau}{2} W) V_\perp.\)
   3.4. \(\tau_{k+1} \) perform BB update.

4. until \(\|l(k) - l(k-1)\|_\infty\) is small or \(k > k_{max}\)

The computations involved in the above manifold optimization algorithm, however, raise questions over its scalability to high-dimensional data. Also, the presence of too many free parameters (including those for the BB update) increases the model complexity as well as its sensitivity to fine-tuning. Although the algorithm can minimize any smooth function over the Stiefel Manifold, the special structure of \(l(V_\perp)\) in (17) calls for a rather simple yet effective method. If we substitute \((X - O)\) by \(Q\) and \((1\mu T + S)\) by \(R,\) it is interesting to note that the optimization looks similar to the orthogonal procurers rotation (PR) problem [23] as

\[
\min_{V_\perp} \frac{1}{2}\|QV_\perp - R\|^2_T, \quad \text{s.t.} \quad V_\perp^T V_\perp = I.
\]

(20)

But the PR solution cannot be used directly, since \(V_\perp \in \mathbb{R}^{p \times d}\) is not a square orthonormal matrix. On the other hand, we can
plug in an iterative algorithm by alternatively performing a linearized gradient descent and a PR solution to the nearest orthonormal matrix estimation problem.

In terms of $Q$ and $R$, the Euclidean gradient of $l$ over $V_{\perp}$ is $Q^T(QV_{\perp}^{-1} - R)$.

Using the same linearization approach, the iterative estimation equation now looks like

$$ V_{\perp}^{(j+1)} = V_{\perp}^{(j)} - \frac{1}{\|Q\|^2} Q^T(QV_{\perp}^{-1} - R) $$

(21)

where the step size is given by the inverse of the spectral norm of the Hessian, which is $\|Q\|^2$.

The next step is to find the closest orthonormal matrix to $V_{\perp}^{(j+1)}$ so that the Stiefel manifold constraint is satisfied. This can be mathematically written as

$$ \min_W \|W - \tilde{V}_{\perp}^{(j+1)}\|_F^2, \text{ s.t. } W^TW = I. $$

(22)

The Frobenius norm in the above can also be represented as

$$ \|W - \tilde{V}_{\perp}^{(j+1)}\|_F^2 = \text{tr}\left((W - \tilde{V}_{\perp}^{(j+1)})^T(W - \tilde{V}_{\perp}^{(j+1)})\right). $$

(23)

**Algorithm 4: ($V_{\perp}$)-Opt Lin+PR**

**Input:** $Q, R$

**Output:** $V$

1. Initialize $V^{(0)}_{\perp} \in \mathbb{R}^{p \times d}$ as an orthonormal matrix.
2. $j = 0, \Delta V^{(0)}_{\perp} = 0$.
3. repeat
   1. $j = j + 1$.
   2. $\Delta V_{\perp}^{(j)} = \frac{1}{\|Q\|^2} Q^T(QV_{\perp}^{-1} - R)$.
   3. $T = V_{\perp}^{(j-1)} - \Delta V_{\perp}^{(j)} + a\Delta V_{\perp}^{(j-1)}$.
   4. Do SVD of $T = U_W \Sigma_W V_W^T$.
   5. $U_{\text{econ}}^W$ = first $d$ columns of $U_W$.
   6. $V_{\perp}^{(j)} = U_{\text{econ}}^W V_W^T$.
4. until $j > j_{\text{max}}$

On decomposing the product inside the trace operator and using $W^TW = I$, the only term left that still depends on $W$ is $\text{tr}(W^T \tilde{V}_{\perp}^{(j+1)})$. Also, the negative sign in front of this term changes (22) to a trace maximization problem. The new optimization is given by

$$ \max_W \text{tr}(W^T \tilde{V}_{\perp}^{(j+1)}), \text{ s.t. } W^TW = I. $$

(24)

The singular value decomposition (SVD) of $\tilde{V}_{\perp}^{(j+1)}$ gives

$$ \tilde{V}_{\perp}^{(j+1)} = U_W \Sigma_W V_W^T $$

(25)

where $\Sigma_W = [\text{diag}(\sigma_1 \ldots \sigma_d); (p-d) \times d]$ is the horizontal concatenation of a diagonal matrix of singular values and an all-zero matrix. Also, $U_W$ and $V_W$ are orthogonal matrices in $\mathbb{R}^{p \times p}$ and $\mathbb{R}^{d \times d}$, respectively. The SVD, in its economy size version, can also be written as

$$ \tilde{V}_{\perp}^{(j+1)} = U_{\text{econ}} \Sigma_{\text{econ}} V_{\text{econ}}^T $$

(26)

where $\Sigma_{\text{econ}}$ is just a diagonal matrix with its diagonal entries being $(\sigma_1 \ldots \sigma_d)$ and $U_{\text{econ}} \in \mathbb{R}^{p \times d}$ refers to the $d$ columns of $U_W$ corresponding to the nonzero rows of $\Sigma_W$.

We define an $\mathbb{R}^{d \times d}$ orthogonal matrix as $Z = V_W^T W^T$.

Now

$$ \text{tr}(W^T \tilde{V}_{\perp}^{(j+1)}) = \text{tr}(W^T U_{\text{econ}} \Sigma_{\text{econ}} V_W^T). $$

(27)

The right-hand side in (27) can also be written as $\text{tr}(Z \Sigma_W)$ which can be further represented as

$$ \text{tr}(Z \Sigma_W) = \sum_{i=1}^{d} \sigma_i Z_{ii} \leq \sum_{i=1}^{d} \sigma_i $$

(28)

where the last inequality is based on von-Neumann’s trace inequality [27] on the product of two square $d \times d$ matrices and the fact that the singular values of an orthogonal matrix $Z$ are all 1. Given that $\sigma_i$ are all nonnegative, the upperbound can be attained by letting $Z$ to be equal to $I_{d \times d}$. Thus, the best approximation of $W$ is given by $U_{\text{econ}}^T V_W^T$. Therefore, the estimate of the OC space at the end of the next iteration is

$$ V_{\perp}^{(j+1)} = U_{\text{econ}}^T V_W^T. $$

(29)

The $V_{\perp}$-optimization problem may contain multiple local minima of interest. In order to speed up the convergence, we make use of accelerated gradient methods like the momentum technique which is popularly used in training neural networks [13]. The update of $V_{\perp}$ in the previous iteration is added to that of the current iteration after multiplying by a factor $\alpha \leq 1$. This revised update is now used in Algorithm 4.

There are also other speedup techniques, such as Nesterov’s accelerated gradient and AdaGrad [24].

We claim that the procedure followed in Algorithm 4 guarantees that the subspace orthogonality constraint is satisfied and the loss function is nonincreasing during the iteration: $l(V_{\perp}^{(j+1)}) \leq l(V_{\perp}^{(j)})$. In fact, this can be proved by defining a surrogate function

$$ g(\Xi, V_{\perp}) = l(V_{\perp}) + (\nabla l(V_{\perp}), \Xi - V_{\perp}) + \frac{\rho}{2} \|\Xi - V_{\perp}\|^2_F. $$

(30)

Based on Taylor expansion

$$ g(\Xi, V_{\perp}) - l(\Xi) = \frac{\rho}{2} \|\Xi - V_{\perp}\|^2_F - l(\Xi) - l(V_{\perp}) - (\nabla l(V_{\perp}), \Xi - V_{\perp}). $$

(31)

Given $V_{\perp}$, the problem of $\min_{\Xi} g(\Xi, V_{\perp})$ reduces to

$$ \min_{\Xi} \Xi - \left(V_{\perp} - \frac{1}{\rho} \nabla l(V_{\perp})\right) \|\Xi - V_{\perp}\|^2_F, \text{ s.t. } \Xi^T \Xi = I. $$

(32)

Therefore, it is easy to see that the inequality

$$ l(V_{\perp}^{(j+1)}) \leq g(V_{\perp}^{(j+1)}, V_{\perp}^{(j)}) \leq g(V_{\perp}^{(j)}, V_{\perp}^{(j)}) = l(V_{\perp}^{(j)}) $$

(33)

holds if $\rho \geq \|Q\|^2_2$.

### C. Batch Version

Algorithm 4, with its computational burden, is not fast enough to deal with usually high-dimensional ($p > n$) real-world data. For example, in case of camera images, the dimensionality $p$ is the number of pixels which can be of the order of thousands or millions. The intrinsic dimensionality
r is usually very small making d of the same order as p. However, thanks to the iterative nature of the R²-PCP design, the columns of the OC subspace projection can be estimated in batches. This reduces the dimensionality of the problem in each subproblem and a clever technique can combine the estimates to give the final V ⊥.

The plain version tries to solve (20), where V ⊥ has p rows and d columns. The columns of V ⊥ can be grouped into batches of size mi each, where i varies from 1 until k and \( \sum_{i=1}^{k} m_i = d \). Similarly, the columns of R can also be grouped into submatrices \( \mathbf{R}_i \) defined by the same batch sequence \( m_i \). So, now effectively we have by concatenation \( V_\perp = [V_{\perp 1}^T \ V_{\perp 2}^T \ \ldots \ V_{\perp i}^T]^T \) and \( R = [\mathbf{R}_1 \ \mathbf{R}_2 \ \ldots \ \mathbf{R}_k] \). Since the objective function in (20) is the square of the Frobenius norm, it can now be represented as

\[
\|QV_\perp - R\|_F^2 = \sum_{i=1}^{k} \|QV_{\perp i} - R_i\|_F^2. 
\]  

It is only the projection space that matters and not the way the spanning vectors are calculated. In order to reach a local minima solution, the optimization can be approximated as solving k subproblems such as

\[
\min_{V_{\perp i}} \frac{1}{2} \|QV_{\perp i} - R_i\|_F^2, \quad \text{s.t.} \ V_{\perp i}^T V_{\perp i} = I 
\]  

\( \forall i = 1, 2, \ldots k \). This considerably reduces the search space of each optimization and can help achieve a local minimum quickly. There is of course a tradeoff between k and the accuracy of estimation. Algorithm 4 computes multiple SVDs of large matrices at a computational cost of \( O(pd^2) \). Assuming the expected value of the \( m_i \) values to be equal to \( m \), the new computational complexity of the batch version gets reduced to \( O(kpm^2) \), or also equal to \( O(pdm) \). For high-dimensional data where \( d \) is sufficiently high, this batch version plays a significant role in fastening the estimation process.

Algorithm 5: (Batch V ⊥)-Opt Version 1

Input: \( X, \ O, \mu, S, [m_1, m_2, \ldots m_k] \)

Output: \( V_\perp \)

1. \( j = 0 \)
2. \( Q = (X - O) \)
3. \( R = [\mu^T + S] \)
4. \( s = 1 \)
5. \( SS = 0 \)
6. \( V_\perp = NULL \)

while \( i < k \)

1. Form \( \mathbf{R}_i \) by selecting \( m_i \) columns of \( R \) starting from the \( s \)th columns.
2. Call \( V_{\perp i} = V_\perp \text{-opt}(Q, \mathbf{R}_i) \)
3. \( Z = (I - SS) V_{\perp i}^T \)
4. Concatenate \( Z \) to the right of \( V_\perp \).
5. \( SS = SS + V_{\perp i} V_{\perp i}^T \)
6. \( s = s + m_i \)

until \( i > k \)

However, \( V_\perp \) formed by concatenating these individual solutions for \( V_{\perp j} \) may not generate an orthonormal set of bases for the OC subspace. Although \( V_{\perp j} \) and \( V_{\perp j} \) are separately orthonormal for distinct \( i \) and \( j \), they are not orthogonal to each other \( V_{\perp j}^T V_{\perp j} \neq 0 \). Applying the Grah–Schmidt orthogonalization on the concatenated matrix \( [V_{\perp 1} \ V_{\perp 2} \ldots V_{\perp i}] \) will let the computational complexity be \( O(pd^2) \). However, we can take advantage of the special sequential structure of the individual \( V_{\perp j} \)'s. The first estimate \( V_{\perp 1} \) is itself orthogonal and thus can be left untouched. Let \( V_{\perp 1} = V_{\perp 1} \). The second estimate \( V_{\perp 2} \) can be orthogonalized with respect to \( V_{\perp 1} \) by

\[
V_{\perp 2} = V_{\perp 2} - V_{\perp 1} (V_{\perp 1}^T V_{\perp 2}) 
\]  

which can also be written as

\[
V_{\perp 2} = (I - V_{\perp 1} V_{\perp 1}^T) V_{\perp 2}. 
\]  

We can now horizontally concatenate \( V_{\perp 1} \) and \( V_{\perp 2} \) to form an orthonormal matrix with \( m_1 + m_2 \) columns. Similarly, \( V_{\perp 3} \) can now be orthogonalized with respect to this concatenated matrix

\[
V_{\perp 3} = (I - (V_{\perp 1} V_{\perp 1}^T + V_{\perp 2} V_{\perp 2}^T) V_{\perp 3}. 
\]  

However, \( V_{\perp 1} V_{\perp 1}^T \) has been precomputed. Thus, at each \( i \)th iteration, we just have to compute \( V_{\perp i} V_{\perp i}^T \) and add it to the previous sum \( SS \) in the equation

\[
V_{\perp i} = (I - (SS + V_{\perp i-1} V_{\perp i-1}^T)) V_{\perp i}. 
\]  

This considerably reduces the time of computing the orthonormal OC subspace and feeding it to \( \mu-O-S\)-opt alternatively. Algorithm 5 details the overall computation.

Algorithm 6: (Batch V ⊥)-Opt Version 2

Input: \( X, m_i \)

Output: \( V_\perp, \ O, \mu, S \)

1. \( j = 0 \)
2. \( m_0 = 0 \)
3. \( \sum_{i}^{j-1} m_i = s \)
4. Call \( V_{\perp 1} \)-opt on \( X - O^{(j)} \) to get \( V_{\perp 1} \in \mathbb{R}^{(p-s) \times m_1} \).
5. Call \( (\mu, O, S)\)-opt on \( X \) but with dimension of \( \mu \) and \( S \) being \( m_1 \).
6. Find its orthogonal complement \( V_{\perp 1}^\perp \) which is of \( p - s - m_1 \)-dimensions.
7. Deflate \( X - O^{(j+1)} = X - O^{(j)} V_{\perp 1}^\perp \).
8. \( j = j + 1 \)
9. until \( j \geq k \)
10. Overall PC subspace \( V = \prod V_{\perp i} \).
11. And OC subspace estimation \( V_{\perp} = \text{null}(VV^T) \).
12. Call \( (\mu, O, S)\)-opt one final time using \( V_{\perp} \).

Another version of computing in batches is to have an iterative-estimation-and-deflation style [37] computing of the subspace vectors. This batch version of \( R^2\)-PCP divides the work of the overall algorithm by alternatively calling \( V_{\perp j}\)-opt and \( (\mu, O, S)\)-opt iteratively on the \( p \)-dimensional data \( X \).
only to estimate a partial subspace in each epoch. The final output is a $V_\perp$ that belongs to $\mathbb{R}^{p \times d}$. But it reduces the computational burden by dividing the $d$ into several $m_i$ values, such that $\sum_{i=1}^{k} m_i = d$. Each time we have an estimate for a partial rank-$m_i$ OC subspace, we deflate the design matrix accordingly. This ensures that the next set of estimated subspace vectors is independent of the ones estimated in the previous iteration. Though we have to run the $R^2$-PCP $k$-times, each problem size is now much smaller and we can also have an early termination criterion for iterations, which can speed up the overall computation. Both the batch versions have been tested successfully in reducing the computational burden of the subspace estimation algorithm. It is up to the choice of the designer to decide between the two proposed techniques.

V. EXPERIMENTS

A. Experiments on Simulation Data

In this section, we test the performance of $R^2$-PCP against other robust subspace estimation methods on simulated data with e-Type $O$ and r-Type $S$ outliers. The techniques considered for comparison are RAPCA [29], ROBPCA [28], LRR [38], Go Decomposition (GoDec) [30], ROSL [32], and PCR [3] in its noise-considered stable version. In total, $n$ observations of $p$-dimensions are generated with only $r = 3$ being the dimension of the true PC subspace. Thus, only $r$ is relevant and enough in explaining the true data. The data varies along the three principal directions with $\Sigma = \text{diag}(64, 36, 16)$ and $\mu = 0$. It is now added with zero-mean Gaussian noise. For our experiments, we keep the noise variance as $\sigma^2 = 0.5$. A percent variable $\psi$ is introduced to denote the percentage or sparsity of outlying entities. Since we focus here only on the e-Type $O$ and r-Type $S$ combination, $(\psi/2)$ fraction of the rows are outliers in the OC subspace, and additionally, $(\psi/2)$ fraction of the total elements in the $n \times p$ design matrix $X$ are large-valued garbage entries. Three different choices of $\psi \in \{4\%, 10\%, 16\%\}$ are experimented with and the magnitude of the outliers is taken to be large enough, say $L = 10$. This setting is used for creating a sparse matrix $S$ (with r-Type outliers) which has $n \times \psi$ of its rows as $[L, L, \ldots, L]$ and 0 otherwise. The sparse matrix $O$ (with e-Type outliers) has all 0 values except for the value of $L$ in 0.5 $n \times p \times \psi$ of its randomly selected positions. Experiments are conducted for the low-dimensional case ($n > p$) with $n = 50$ and $p = 15$, and the high-dimensional case ($n < p$) with $n = 50$ and $p = 100$. The PC subspace estimation accuracy is measured in terms of cosine affinity (the cosine value of the largest canonical angle between the original PC subspace and the estimated PC subspace) and is averaged over 50 independent simulations and reported in Tables I–III. A value of 1 indicates perfect estimation.

For quantifying the accuracy of outlier detection, three benchmark measures are used: the mean masking (M) and swamping (S) probabilities, and the rate of successful joint detection (JD). While masking probability is the fraction of undetected outliers, the swamping probability is the fraction of good points labeled as outliers. JD is the fraction of simulations with zero masking. An ideal method should have $M = 0$, $S = 0$, and $JD = 1$. In our case, masking is a much more serious problem than swamping, because the former can cause serious distortion in estimating true subspace. Thus, the upper bound parameters for outlier numbers are set to be $q_S = \lceil0.75 \times n\times \psi\rceil$ and $q_O = \lceil0.75 \times n \times p \times \psi\rceil$. Both correspond to approximately 1.5 times of the true number of outliers. Also, we ignore those e-Type outlier entities in matrix $O$ if the corresponding rows are detected as OC outliers (nonzero rows of $S$). The performance of $R^2$-PCP, in terms of the above measures, is reported in Table IV.

As can be seen from the results, $R^2$-PCP clearly edges over PCP and its other robust variants when it comes to how close the estimated subspace is to the true PC subspace in presence of both OS and OC outliers. The plain PCA, as expected, fails drastically. It is interesting to note that PCP performs as good as $R^2$-PCP when there are very few outliers, like $\psi = 4\%$. However, when the percentage of outliers increases, the performance of other algorithms deteriorates drastically as compared with $R^2$-PCP. Also, in the high-dimensional cases, our algorithm has an even bigger margin over the rest of the techniques. Only GoDec is as consistent but still its cosine affinity of estimated subspace is poorer than that of
they are totally independent of the factors that defines the real person’s face images. The rest 31 face images are assumed to be corrupted by an additive sparse outlier matrix in the OS (e-Type $O$), which refers to the cast shadows and specularities. In running the $R^2$-PCP algorithm, we set $r = 9$, $q_S = 4$, and $q^e_O = 0.04 \times 35 \times 896 = 1250$, due to the sparsity assumption of the outliers and the insensitivity property of upper bound of the number of outliers.

Plotting eigenfaces is a way of visualizing the PC vectors or columns of the estimated subspace $V$ by reordering them back into matrices of the size of original images. Fig. 6 shows eigenfaces formed by reshaping the vectors of the principal subspace estimated by PCP into the shape of original images in the top row. Similar eigenfaces generated by $R^2$-PCP are shown in the bottom row. It can be seen that a couple of the eigendimensions in the case of PCP look very noisy and may have absolutely nothing informative to describe the true subspace. In each row of Fig. 7, from left to right, we show in order the original face image, the detected outliers in the OS (white pixels represent nonzero entries in $O$) and the low-rank approximation $B$ of the face images obtained by applying $R^2$-PCP. Clearly, the irregular pixels, like along the nose bridge, the forehead and the nose tip are detected as outliers. $R^2$-PCP can well compensate for these and provide good face recovery. Also, using our algorithm, we could perfectly identify the locations of the nonface images as OC outliers.

Face recognition is an important aspect of profile learning in online social network portals. A user may post pictures of scenic wallpapers, food, cartoons, beaches, or other kinds of images along with those containing his own face. Also, the face pictures may be marred by various elementwise OS space outliers. In this case, $R^2$-PCP is a one-shot method to identify the subspace by eliminating both kinds of outliers. About 280 training face images from the Yale Database belonging to 8 different persons, labeled 1 to 8, and some 20 real-world nonface images which are pictures of flowers, beaches, and cartoons. These outliers were also allotted a label randomly selected from 1 to 8. A sample set of the pictures has already been shown in Fig. 3. A 9-D subspace was estimated using PCA, robust PCA, and $R^2$-PCP and the data were projected onto this subspace as an act of feature transformation or dimensionality reduction. It is assumed that if the true subspace can be perfectly estimated, projection of the data points onto it can make the different classes more separable. The test set consisted of 224 images of those 8 persons and the estimated projection space is also applied on them. The dimension reduced data were then fed to a linear classifier. Not only could $R^2$-PCP estimate a better subspace, but also detect

<table>
<thead>
<tr>
<th>$\sigma_2$</th>
<th>$\psi$</th>
<th>M</th>
<th>S</th>
<th>JD</th>
</tr>
</thead>
<tbody>
<tr>
<td>4%</td>
<td>0.015</td>
<td>0.016</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>5%</td>
<td>0.001</td>
<td>0.040</td>
<td>0.0975</td>
<td></td>
</tr>
<tr>
<td>15%</td>
<td>0.005</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16%</td>
<td>0.000</td>
<td>0.016</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>0.000</td>
<td>0.038</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>60%</td>
<td>0.000</td>
<td>0.050</td>
<td>1.000</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. Different Faces of a single person together with random outlying images.

Fig. 6. Top row: eigenfaces obtained using PCP. Bottom row: $R^2$-PCP.
TABLE V
MEAN AND STANDARD DEVIATION (STD.) OF MISCLASSIFICATION PERCENTAGE ERROR IN FACE RECOGNITION IN PRESENCE OF BOTH TYPES OF OUTLIERS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean Misclassification</th>
<th>Std. of Misclassification</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>26.69</td>
<td>1.79</td>
</tr>
<tr>
<td>PCP</td>
<td>23.52</td>
<td>2.77</td>
</tr>
<tr>
<td>R²-PCP</td>
<td>17.41</td>
<td>1.26</td>
</tr>
</tbody>
</table>

the presence of outliers and subsequently remove them from subsequent learning. The rows in the projection of training data onto the subspace which correspond to the estimated rowwise outliers were removed while training the linear classifier in case of R²-PCP. In Table V, it can be seen that the misclassification percentage in the case of R²-PCP is considerably lower than PCA or PCP. This advantage can be attributed to its efficiency in learning a more accurate subspace along with removing unwanted rowwise and elementwise outliers reliably.

C. Video Background Modeling
Finally, we also applied our algorithm in video background subtraction. A video sequence of a scene where the background is still and only few objects are mobile can be considered as an addition of a low-rank background video and an outlier video that contains just the moving objects. We used the popular hall video data set [18] for our experiments. However, we added another moving object, which can either be a bus, train, bird, or any obstruction in front of the video camera, that is either large or close enough to block the camera’s sight of the true background for at least a few frames. Thus in the current setup, the background information is completely lost in a few frames of the video. Such cases have a high probability of occurrence in real life situations. If a robust PCA algorithm fails to detect these frames as whole example outliers, the estimated background may not be the true fixed background and we may also have discrepancies in object tracking. In our experiments, this obstruction was a randomly
In this paper, we have proposed a novel design model for robust subspace learning that can deal with outliers both in the observation and OC space separately. Algorithms were developed to iteratively estimate the true subspace and outlying matrices. Experiments on artificially generated data clearly proved its efficiency over other robust PCA models. We were also able to compute a low-dimensional subspace of human face images and classification results on projected data indicate that the estimated subspace is more relevant to the learning task as compared to that extracted using PCA or PCP. R^2-PCP could identify the nonface images perfectly as r-type OC outliers. It was also successful in pointing out specular reflections and cast shadows as e-type OS outliers. Our model also finds semantic application in video surveillance specially when the given background is blocked from the camera’s view for certain amount of time. The recovery of the fixed background and moving objects in the foreground using R^2-PCP was shown to be much better in comparison to simple PCP. The reinforcement in our robust PCA analysis can be attributed to several factors like the usage of nonconvex sparsity-enforcing penalties, treating OC outliers with special care and accommodating whole sample rejection along with just elementwise anomalies. Although this efficiency is attained at a relatively higher cost of computation, we have outlined a few batch-iterative faster versions of the V⊥-optimization. A detailed analysis of these batch algorithms on big data sets is yet to be done and is considered as one of our future works. There have been various modifications and improvements upon
PCP in the literature and we believe that combining those changes with $\mathbb{R}^2$-PCP can lead to even powerful subspace learning methods. Such an approach can also be extended to nonlinear subspace representation algorithms like manifold learning and especially deep neural networks which have been lately very successful in various applications. It has been shown that generative deep neural networks \cite{34} can flatten the highly entangled raw data manifolds and help capture the governing low-dimensional subspace. Considering the OC space outliers in such architectures can also provide interesting insights and results.

REFERENCES

\cite{2} E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” J. ACM, vol. 58, no. 3, p. 11, May 2011.
\cite{31} Y. She, S. Li, and D. Wu, “Robust orthogonal complement principal component analysis,” J. Amer. Statist. Assoc., to be published.

Pratik Prabhanjan Brahma, photograph and biography not available at the time of publication.

Yiyuan She, photograph and biography not available at the time of publication.

Shijie Li, photograph and biography not available at the time of publication.

Jiade Li, photograph and biography not available at the time of publication.

Dapeng Wu, photograph and biography not available at the time of publication.

AQ:6
AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your paper. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

AQ:1 = Please confirm whether the edits made in the financial information are OK.
AQ:2 = Please indicate the section numbers for the phrase “the later sections” throughout the article.
AQ:3 = Figs. 5–7 have been rearranged to ensure sequential order. Please confirm.
AQ:4 = Please provide the issue no. or month for ref. [12].
AQ:5 = Please provide the volume no., issue no., page range, and year for ref. [31].
AQ:6 = Please provide authors’ biographies.
Reinforced Robust Principal Component Pursuit
Pratik Prabhanjan Brahma, Yiyuan She, Shijie Li, Jiade Li, and Dapeng Wu, Fellow, IEEE

Abstract—High-dimensional data present in the real world is often corrupted by noise and gross outliers. Principal component analysis (PCA) fails to learn the true low-dimensional subspace in such cases. This is the reason why robust versions of PCA, which put a penalty on arbitrarily large outlying entries, are preferred to perform dimension reduction. In this paper, we argue that it is necessary to study the presence of outliers not only in the observed data matrix but also in the orthogonal complement subspace of the authentic principal subspace. In fact, the latter can seriously skew the estimation of the principal components. A reinforced robustification of principal component pursuit is designed in order to cater to the problem of finding out both types of outliers and eliminate their influence on the final subspace estimation. Simulation results under different design situations clearly show the superiority of our proposed method as compared with other popular implementations of robust PCA. This paper also showcases possible applications of our method in critically tough scenarios of face recognition and video background subtraction. Along with approximating a usable low-dimensional subspace from real-world data sets, the technique can capture semantically meaningful outliers.

Index Terms—Manifold optimization, nonconvex penalties, robust principal component analysis (PCA), subspace learning.

I. INTRODUCTION

THERE has been a huge amount of research done with the objective of recovering low-dimensional structures from high-dimensional data, such as images, videos, text documents, and bioinformatics. Due to the curse of dimensionality and algorithm scaling issues, it is important to estimate a low-dimensional representation of the data which can ease the subsequent learning tasks. Classical principal component analysis (PCA) [1] tries to find out the directions of maximum variation and is undoubtedly the most popular method to serve the purpose. PCA can also be framed as a low-rank approximation problem [2]. However, due to the \(\ell_2\) loss used in the objective function of PCA, it becomes impractical and ineffective when the observations are grossly corrupted with outliers and noise.

To robustify PCA, [2] proposed a new model assuming that the design matrix \(X \in \mathbb{R}^{n \times p}\) of \(p\)-dimensional variables can be written in the form of \(X = B + O\), where \(B\) is the low-rank matrix approximation with \(\text{rank}(B) \leq r\) and \(O\) is the sparse outlier matrix with few large-valued entries in random positions. The joint recovery of \(B\) and \(O\) can be characterized by minimizing the objective function \(\text{rank}(B) + \lambda \|O\|_0\), where \(\|\cdot\|_0\) denotes the elementwise \(\ell_0\) norm which is the number of all nonzero entries in the matrix. In order to facilitate the computation and analysis, [2] provides a convex relaxation of the formulation called principal component pursuit (PCP) as

\[
\min_{B, O} \|B\|_* + \lambda \|O\|_1 \quad \text{s.t.} \quad X = B + O
\]

where \(\|\cdot\|_*\) denotes the matrix nuclear norm (sum of all singular values), and \(\|\cdot\|_1\) denotes the elementwise \(\ell_1\) norm (sum of the magnitudes of all matrix entries). Following its huge success in robust subspace estimation, there have been various extensions of PCP, such as [3]–[5]. An exhaustive survey of robust PCA methods is provided in [7]. Such models have found widespread applications, particularly in image and video analysis like learning the invariant subspace in face recognition [6] and subtracting fixed low-rank background from video to do object tracking [8], [9].

However, PCP and other techniques built on top of it face a lot of limitations. First, the \(\ell_1\) penalty applied on the sparse outlier matrix is only the best convex approximation to the sparsity-enforcing \(\ell_0\) norm. Though popularly used in lasso type of problems, it cannot handle collinearity and may end up in inconsistent selection according to the irrepresentable conditions problem [10]. It also introduces unnecessarily extra bias in the final estimation. This has been the motivation behind switching to nonconvex penalties, such as SCAD [11] and \(\ell_0 + \ell_2\) [12]. Second, it has been noticed that the tuning of the parameter \(\lambda\) is difficult and requires experimental expertise, since it does not speak directly of the number of outliers in the given data. Instead of using a Lagrange multiplier, it is perhaps more suitable to apply direct threshold in form of constraining inequalities on the number of outlying elements that makes more sense from a practitioner’s perspective. This also corresponds to the popularly used variable screening techniques [14] where only a set of features are chosen based on selection rules.

But the inability of PCP to deal with certain types of outliers is more serious than the above two issues. Although PCP can effectively identify outliers in the original observation space (OS) as an additive component matrix, it may fail in the presence of potential outliers in the orthogonal complement (OC) subspace, which is the space orthogonal to the authentic PC subspace. This paper shows that any algorithm with the aim of robust subspace learning should specifically identify and remove the outliers in OC subspace, because these...
can skew the subspace estimation drastically. While PCP and its variants treat all of the outliers from the perspective of the OS, there may exist some OC outliers which go unnoticed as good observations.

Section II raises this concern and explains the various types of situations with outliers and their effects using both simulation and real-world data. Section III attempts to come up with a mathematical model to achieve the estimation of the principal subspace in presence of all kinds of outliers. A set of algorithms are developed in Section IV to solve a truly robust version of PCA, which is named the reinforced robust PCP (R²-PCP). The computational algorithms are then used in Section V on simulation data to show the performance advantages of R²-PCP against various other methods for robust PCA. The algorithms are also tested on machine learning tasks such as face recognition and video background subtraction to show improvement over classical PCA and robust PCP. Finally, we conclude this paper by summarizing the various results and remarks observed from the different experiments.

II. WHAT ARE OUTLIERS

Outliers are arbitrarily large-valued measurements, which do not characterize the true data samples. While the unobservable noise is assumed to have a fast-decaying Gaussian probability distribution, outliers are observations or values that are considerably different from the majority of the data and usually follow heavy-tailed distributions. If the observed examples are stacked as rows in data matrix as shown in Fig. 1, either some of the particular entries may be affected due to additive outliers or entire rows (whole examples) are corrupted. On the basis of location of the outliers in the data matrix, outliers can be classified into two types.

1) r-Type Outliers: Those observations or examples which are wholly corrupted and do not belong to the true principal data subspace are termed rowwise (r-type) outliers.

2) e-Type Outliers: When some of the entries, but not all, at arbitrary locations in the design matrix have outlying values, they are considered to be elementwise (e-type) outliers.

As will be shown later, PCP performs poorly when there are too many r-type outliers in the data. Also, the outliers may not be always visible in the observed design matrix. A deeper analysis of the different types of outliers is shown in Fig. 2. The true data in the observable R³ space is actually a finite linear R¹ manifold or just a straight line. This direction, along which the data varies the maximum, is also the PC subspace. Our objective is to accurately identify this subspace in presence of noise and outliers.

Point O is the data center and, for esthetic reasons, it is taken to be the origin itself. If an outlier lies along the PC, it does not adversely affect the PC subspace estimation. The subspace estimation is largely skewed when there are outliers (or outlying components) away from the PC subspace. The directions orthogonal to the PC subspace span the OC subspace. In Fig. 2, it is the R² plane to which the true data line acts as a normal. Data points like A, with large enough OC projection components, are the ones that cause maximum damage to the performance of any robust PCA technique. A challenge is that in the OS, their coordinates may lie within the range of variation in actual data. Therefore, these outliers may appear as good observations. The big question is whether PCP and its variants can remove the influence of such outliers from the estimation of the true subspace. Our simulation results in the later sections disclose the weakness of these algorithms in handling such cases. We also go ahead to argue that such outliers can practically exist in real-world data and it is appropriate to deal separately with these OC outliers after transforming the OS properly.

A taxonomy that can ease the handling of outliers in our subsequent mathematical modeling is considered here. While the previous classification was on the basis of location in the matrix, the following classification considers the subspace in which these are most prominent. The types of outliers in a robust PCA setup can be classified into two types.

1) OS Outliers: Those entries in the data matrix which have relatively anomalous values as compared to the range of variation in the true data are considered to be OS outliers.

2) OC Outliers: The observations which have outlying magnitude when projected onto the OC subspace and
which have also not been treated as visible outliers in
the OS are considered to be OC outliers.

In our subsequent analysis, OC outliers are usually taken
to be the whole observation itself and hence are r-type in
nature. Data point A in Fig. 2 is a good example of an
OC outlier. It is interesting to note that if we project all
the data points onto the OC subspace, the true data samples
capture a small volume around the data center O. However,
the projection of point A has a large row-norm or \( \ell_2 \) Euclid-
ean distance from the origin O. Even if the coordinates of
A may lie within the range of variation of the true data
samples, its row-norm in OC subspace can still be large
enough, especially in high-dimensional scenarios. This makes
the whole sample a prominent outlier that is better disregarded
from consideration as it is common with the robust statistics
literature.

Such a branching of outlier types makes it convenient
to handle these observations independently. For example,
in Fig. 2, point A is a r-type OC outlier, while point B
can be e-type OS outliers. The PCP type of algorithms
assumes just additive outliers \( O \) with large values in the
OS, as shown in (1). However, in this paper, along with the
OS outlier matrix \( O \), we design a more complete model by
including a sparse matrix \( S \) that specifically considers outliers
in the OC subspace. Thus, r-type OC outliers correspond
to row sparsity of \( S \). On being projected back to the OS
for modeling purposes, we come up with the decomposition
\( X = B + O + (1 \mu^T + S)Y + E \), where \( \mu \) is the data center
vector and \( E \) is additive noise. Section III describes the
mathematical model in detail. For all our subsequent analysis,
we go with the combination of r-type OC and e-type OS
outliers because of its semantic implications in real-world
situations. However, other combinations can also be easily
handled by our design framework.

Computer vision problems involve high-dimensional
(according to the number of pixels and channels, such as
R, G, B, and depth) image data sets, which usually belong
to low-dimensional manifolds. Hence, subspace estimation
can help in reducing the dimensionality of the problem and
easing subsequent learning tasks. However, these data sets are
frequently filled with unnecessary inclusions, which make the
estimation difficult. For example, recognition of humans from
camera images of their faces may suffer from outliers, such as
glasses, beard, or other kinds of occlusions distorting the true
facial appearance of the person. In the presence of varying
illumination, cast shadows and specular reflections should
also be ignored while learning the actual face subspace. In
perspective of the data matrix \( X \) formed by putting all the
pixel intensity values of a single image in each row, these
unwanted occlusions are e-type outliers as only some of the
pixels are affected. With the recent surge of online social
networks, recognizing pictures requires a robust training of
invariant features defining an individual’s face. Often the
picture folders of users, like in Fig. 3, contain face images
along with nonface images, such as scenic wallpapers, food,
cartoons, and beaches. Also, some of the face pictures, which
are heavily jittered or blurred, may hinder the estimation of
the true subspace. These images are whole examples which
are also r-type outliers and should be removed while learning
the principal subspace.

While PCP would treat all of these outliers as additive
aberrations in the OS, a fully robust PCA model should
take care of both OS and OC outliers present either as r or
e-type separately. Reflections, shadows, and occlusions affect
certain pixels only and are clearly e-type outliers in the OS,
whereas the nonface images are r-type OC outliers. The set of
hidden factors that describe the images of only the clean faces
does not characterize the generation of the nonface images.
In other words, the OC outliers are independent of the low-
dimensional intrinsic face manifold. Hence, such data points
should show prominent components in the space orthogonal
to true PC subspace. It can be the case that some of them,
like the cartoon in Fig. 3, may seem like genuine face images
and thus may have similar features and patterns. However,
when the entire data set is projected onto the appropriate OC
subspace, the nonface images will lie far away from the small
volume occupied by the clean face data. Thus, the advantage
of treating these as OC outliers is that they are more clearly
distinguishable in the OC subspace as compared with the OS.
This hypothesis is later verified in Section V.

Another problem where robust subspace estimation finds
a suitable application is tracking moving objects for video
surveillance [6]. The static background is usually considered to
be a linear low-rank subspace for a fixed camera, while moving
objects in the video frames are modeled as outliers. A low-
rank matrix approximation of the entire video frame sequence
using PCP or any other robust PCA method can estimate the
fixed background, subtracting which can help track mobile
objects as outliers. Also for nonstationary cameras on mobile
devices, trajectory analysis as in [15] can help design a
more generalized variant of robust PCA for subtracting the
fixed background, which is again a low-rank component in
the \( R^3 \) space, that is with an intrinsic rank less than or equal
to 3. Experiments, however, show that most of these methods
usually break down in the presence of too many outliers. Also,
there might be special situations of blocking the camera view.
As can be seen in Fig. 4, videos captured from a camera on
top of autonomous cars may contain a large number of moving
objects and it might also be the case that the background view
III. MATHEMATICAL MODELING

The objective function of PCP in (1) tries to robustify PCA by decomposing the data matrix X into a low-rank component B and a sparse outlier component O in the OS. In [3], the model of robust PCP is extended to X = B + O + E by considering a Gaussian noise term E. However, as pointed out in Section II, robust PCP-based approaches may be unable to tackle the OC outliers. While ROC-PCA [31] attempts at modeling the OC outliers, it fails in the presence of both types of outliers. To explicitly characterize the outliers both in the OS and the OC subspace, we describe the complete data model as

\[ X = B + O + (1\mu^T + S)V_\perp^T + E \]

with \( \text{rank}(B) \leq r \) and the row space of B is orthogonal to the column space of orthogonal matrix V_\perp, i.e., B = CV_\perp, where C can be an \( n \times r \) matrix. In (2), the given \( n \times p \) data matrix \( X = [x_1, \ldots, x_n]^T \) is decomposed into several components. Let \( d := p - r \) be the dimensionality of the OC subspace.

Each component can now be explained as follows.

1. \( B \in \mathbb{R}^{n \times p} \) is the rank-\( r \) (\( r < p \)) component.
2. \( O = [o_1, \ldots, o_n]^T = [o_i]^T \in \mathbb{R}^{n \times p} \) represents the outliers detectable in the OS.
3. \( SV_\perp^T \) is the outlier component from the OC subspace projected back onto the OS. Here, \( S = [s_1, \ldots, s_n]^T \) is an \( n \times d \) outlier matrix describing the outlyingness of each observation in the OC subspace and \( V_\perp \in \mathbb{O}^{p \times d} \) (where \( \mathbb{O} \) is the Stiefel manifold defined by \( V_\perp^TV_\perp = I \)) characterizes the OC subspace orthogonal to the rank-\( r \) PC subspace. \( \mu \) is a \( d \)-dimensional mean vector for the observations in the OC space.
4. \( E \) is the unobservable noise term which is independent entries sampled from a possible Gaussian \( \mathcal{N}(0, \sigma^2) \) distribution.

Under the Gaussian assumption of the noise, a maximum likelihood (ML) estimation problem can be formed so as to minimize \( \|X - O - B - (1\mu^T + S)V_\perp^T\|_F^2 \). First, \( V \in \mathbb{O}^{p \times r} \) consists of the top \( r \) ideal PC loading vectors of \( X \) and \( [V, V_\perp] \in \mathbb{O}^{p \times r} \) forms a fully orthonormal basis. Since the row space of B is orthogonal to the column space of V_\perp, the low-rank component can also be written as \( B = CV_\perp \), where C can be an \( n \times r \) random matrix. To simplify the mathematical formulation, we consider the decomposition of \( X - O \) into the mutually orthogonal PC and OC subspaces as

\[ X - O = (X - O)V_\perp V_\perp^T + (X - O)S \]

With a little further matrix algebra, the objective function can now be restructured as

\[ \|X - O - B - (1\mu^T + S)V_\perp^T\|_F^2 = \|(X - O)V - C\|_F^2 + \|(X - O)V_\perp - (1\mu^T + S)\|_F^2. \]

The minimization with respect to C makes the first term in (4) \( \|(X - O)V - C\|_F^2 \) always vanish for any \( O \) and \( V \) by setting \( C = (X - O)V \). Therefore, the above ML criterion now reduces to optimally minimizing the second term \( \|(X - O)V_\perp - (1\mu^T + S)\|_F^2 \).

As discussed earlier, the sparsity in \( O \) and \( S \) should also be considered in the optimization. This motivates the use of general sparsity-enforcing constraints to the loss function. A penalized optimization problem is given as

\[ \min_{\mu, O, S, V_\perp} \frac{1}{2} \|(X - O)V_\perp - (1\mu^T + S)\|_F^2 + P_O(O; \lambda_O) + P_S(S; \lambda_S) \]

subject to \( V_\perp^TV_\perp = I \)

where \( P_O(O; \lambda_O) \) and \( P_S(S; \lambda_S) \) are general sparsity penalties with \( \lambda_O \) and \( \lambda_S \) as the regularization parameters, respectively. Let \( \ell(V_\perp, \mu, O, S) = (1/2)\|(X - O)V_\perp - (1\mu^T + S)\|_F^2 \) be the objective function in the (5). Throughout this paper, we refer to the investigation of (5) as \( \mathbb{R}^2 \)-PCP. In addition to offering a robust estimate of the PC subspace \( V \) by isolating outliers in the OS as in PCP, \( \mathbb{R}^2 \)-PCP also aims at being resistant to outliers in the transformed OC subspace. Two main sparsity-enforcing ways, elementwise or group form, can be applied to design \( P_O \) and \( P_S \). All commonly used penalties can be adopted, including \( \ell_1 \), SCAD [11], elastic net [16] “\( \ell_0 + \ell_2 \)” [39], and so on. Since we favor the combination of e-Type \( O \) and r-Type \( S \) for its wide applicability in real data applications, the penalty on the OS outliers \( P_O(O) \) could take the form of

\[ \|O\|_0 \leq q_O^* \]

and that for \( P_S(S) \) is given by

\[ \|S\|_{2,0} \leq q_S^*. \]

The \( \ell_0 \) and group \( \ell_0 \) constraints are used in view of the aforementioned resistance to gross outliers. It is also intuitively easier to tune such threshold parameters instead of dealing with Lagrangian multipliers. A ridge \( \ell_2 \) penalty can also be added to get smooth and regularized estimates. We will discuss how to design nonconvex optimization algorithms to handle various penalties in Section IV.

IV. COMPUTATIONAL ALGORITHM

Our approach to solve the \( \mathbb{R}^2 \)-PCP problem is to alternatively find the estimates of \( (\mu, O, S) \) and \( V_\perp \) and use these to compute each other variables iteratively. This is continued for
A fixed number of iterations or until convergence. The overall solver is given in Algorithm 1. Each of the intermediate steps, i.e., $(\mu, O, S)$-opt and $V_{\perp}$-opt are elaborated in detail in Sections IV-A–IV-C.

**Algorithm 1: R²-PCP Overall Solution**

**Input:** $X, \gamma$

**Output:** $V_{\perp}, \mu, O$ and $S$

Initialize $V_{\perp}(0), \mu(0), O(0)$ and $S(0)$.

$i = 0$.

**repeat**

$i = i + 1$

Using $V_{\perp}(i-1)$,

- do $(\mu, O, S)$-opt.
- Using recently estimated $\mu(i), O(i)$ and $S(i)$, do $V_{\perp}$-optimization.

**until** convergence

A. Optimizing $\mu, O,$ and $S$

With fixed $V_{\perp}$, the optimization reduces to

$$\min_{\mu, O, S} \frac{1}{2} \| XV_{\perp} - 1\mu^T - OV_{\perp} - S \|_F^2 + P_O(O) + P_S(S).$$

Given $O$ and $S$, the $\mu$-optimization is just an ordinary least square problem, since the penalties are independent of $\mu$. The globally optimal solution can be explicitly calculated as $\mu^* = \frac{1}{(n)}(XV_{\perp} - OV_{\perp} - S)'I$. However, optimizing $O$ and $S$ with fixed $\mu$ is relatively challenging. Practitioners may favor different forms of sparsity-enforcing penalties for $P_O$ and $P_S$, including elementwise and groupwise variants. In order to provide a general algorithmic framework to solve such optimization problems, we provide an extended version of the thresholding-shrinkage-based $\Theta$-estimators [33].

A thresholding rule, denoted by $\Theta(\cdot; \lambda)$ with $\lambda$ as a parameter, is defined to be an odd monotone unbounded shrinkage function. A rigorous definition can be found in [12]. In general, it is a set of thresholding rules on the variables we are applying our sparsity constraints on. It can cover all local minimum points and have guaranteed statistical performance [33]. The $\Theta$-estimator is used here to solve group penalized problems with possibly nonorthogonal designs such as

$$\min_{\beta} -L(\beta) + \sum_{m=1}^{M} P_m(\|\beta_m\|_2; \lambda_m)$$

where $L$ is the objective cost function, $\beta_m$ is the $m$th group of the parameter $\beta$ to be estimated, and $P_m$ is penalty function. There is also a universal duality connection between thresholding rules and penalty functions. For example, $P(\beta; \lambda) = \lambda \|\beta\|_2^2$ translates into a simple function

$$\Theta(t; \lambda) = \frac{t}{t + \lambda}. \quad (10)$$

Similarly, the simple function $\Theta(t; \lambda) = t$ if $\|t\| \leq \lambda$ and 0 otherwise translates into the hard-thresholding penalty given by $P(\beta; \lambda) = 1_{\|\beta\| \leq \lambda}(\lambda \|\beta\| - \beta^2/2) + 1_{\|\beta\| > \lambda} \lambda^2/2$.

The analysis in [39] demonstrates that a coordinatewise minimum point of (9) easily be obtained by performing an iterative multivariate thresholding procedure

$$\beta^{(k+1)}_m = \Theta_m(\beta^{(k)} - \alpha \frac{\partial L(\beta)}{\partial \beta_m} ; \lambda_m) \quad (11)$$

Such an iterative procedure is guaranteed to converge if we choose $\alpha$ suitably small enough. This strategy covers all thresholding rules, and all practically used penalties (convex or nonconvex). A special example is the group linear model setup, where $y = X\beta + \epsilon$ with Gaussian noise $\epsilon$, and thus, $-L(\beta) = \|y - X\beta\|^2$. By setting the step size $\alpha$ to be less than or equal to $1/\|X\|_2^2$, where $\| \cdot \|_2$ denotes the spectral norm, the iterations can guarantee convergence by virtue of estimation theory [36].

When different types of outliers occur in $O$ and $S$, we introduce separate thresholding rules. In the following analysis, we consider the recommended $l_0 + l_2$ as an example. To deal with the $r$-Type $O$ and $r$-Type $S$ combination, we take advantage of the elementwise $\ell_0$ constraint ($\|O\|_0 \leq q^r_O$) and the group constraint ($\|S\|_{2,0} \leq q^r_S$) to replace $P_O$ and $P_S$, respectively. Here, $\|S\|_{2,0}$ refers to the number of nonzero rows in $S$. Thus, (8) with fixed $\mu$ becomes

$$\min_{O, S} \ell(V_{\perp}, \mu, O, S) + \eta_O \frac{\|O\|_2^2}{2} + \frac{\eta_S}{2} \|S\|_2^2$$

s.t. $\|O\|_0 \leq q^r_O$ and $\|S\|_{2,0} \leq q^r_S$. \quad (12)

Note that ridge penalties ($\eta_O/2)\|O\|_2^2 + (\eta_S/2)\|S\|_2^2$ are added to deal with collinearity and benefit from the bias-variance tradeoff. Typically, the values of $\eta_O$ and $\eta_S$ can be set small, e.g., $\eta_O = \eta_S = 1e - 3$. This nonconvex fusion penalty usually gives pretty sparse estimates with good statistical accuracy.

To solve for $O$ with $S$ fixed, it suffices to study

$$\min_O \ell(V_{\perp}, \mu, O, S) + \eta_O \frac{\|O\|_2^2}{2} \text{ s.t. } \|O\|_0 \leq q^r_O \quad (13)$$

and similarly with $O$ fixed, (12) becomes

$$\min_S \ell(V_{\perp}, \mu, O, S) + \frac{\eta_S}{2} \|S\|_2^2 \text{ s.t. } \|S\|_{2,0} \leq q^r_S. \quad (14)$$

The gradient of $l = (1/2)\|XV_{\perp} - 1\mu^T - OV_{\perp} - S \|_F^2$ with respect to $O$ and $S$ is computed as $G_O = -(XV_{\perp} - 1\mu^T - S - OV_{\perp})V_{\perp}^T$ and $G_S = -(XV_{\perp} - 1\mu^T - OV_{\perp})S$. The computational algorithm for solving (13) and (14) relies on two nonlinear quantile thresholds $\Theta^#(t; \lambda)$ and $\Theta^#(t; \lambda)$, which are defined as follows:

$$O^{(k+1)} = \Theta^#(O^{(k)} - a_O G^{(k)}; \eta_O; q^r_O) \quad (15)$$

and

$$S = \Theta^#((X - O)V_{\perp} - 1\mu^T; \eta_S; q^r_S). \quad (16)$$

Usually, $a_O = 1/\|V_{\perp}\|_2^2 = 1$ leads to good convergence. Due to the identity design of $S$ in the optimization problem (14), the estimation in (16) gives a globally optimal solution in just one shot. However, $O$ has to be iteratively estimated. We combine the updating of $\mu$, $O$, and $S$ together
Algorithm 2: (\(\mu, O, S\))-Opt for e-Type \(O\) and r-Type \(S\)

\[ k \leftarrow 0. \]

repeat

\[ S^{(k+1)} \leftarrow \tilde{O}^k((X - O^{(k)})V_{\perp} - 1(\mu^{(k)})^T; \eta_S; q_{\tilde{O}}^k). \]

\[ \mu^{(k+1)} \leftarrow \frac{1}{n}((X - O^{(k)})V_{\perp} - S^{(k+1)})^T. \]

\[ j \leftarrow 0; \tilde{O}^{(0)} \leftarrow O^{(k)}. \]

repeat

\[ \tilde{O}^{(j+1)} \leftarrow \tilde{O}^j((I - \alpha_O S\tilde{O}V_{\perp}V_{\perp}^T) + \alpha_O(XV_{\perp} - 1\mu^{(k+1)} - S^{(k+1)})V_{\perp}^T; \eta_O; q_{\tilde{O}}^j). \]

\[ j \leftarrow j + 1. \]

until \(\|\tilde{O}^{(j)} - \tilde{O}^{(j-1)}\|_\infty\) is small

\[ O^{(k)} \leftarrow \tilde{O}^{(j)}. \]

\[ k \leftarrow k + 1. \]

until \(k\) is large enough

and operate alternatively until we obtain decent estimates of all. Algorithm 2 details the entire flow of estimating the concerned variables.

B. Optimizing \(V_{\perp}\) on a Stiefel Manifold

Given \(\mu, O,\) and \(S\), the minimization of \(l\) with respect to \(V_{\perp}\) is equivalent to

\[
\min_{V_{\perp}, \text{s.t. } V_{\perp}^T V_{\perp} = I} l(V_{\perp}) = \frac{1}{2}\| (X - O) V_{\perp} - 1\mu^T - S \|_F^2. \tag{17}
\]

The rank constraint space \(\mathbb{Q}_{p \times d} = \{ V_{\perp} \in \mathbb{R}^{p \times d} : V_{\perp}^T V_{\perp} = I \} \) is geometrically a Stiefel Manifold. A local mimima of our objective function in (17) should be reached while restricting the walk over its surface only. One of the various techniques [17] to solve this is given by the ManiOpt package [19], which attempts at a gradient-based iterative estimate of \(V_{\perp}\).

Here, we adopt a nonmonotone line search scheme together with Barzilai–Borwein (BB) stepsize technique [20]. In comparison with other commonly used inexact line searches, this does not necessarily provide a descent in function value at each step but results in quicker convergence and performs well in large-scale nonlinear optimization. The nonmonotone search scheme performs backtracking only occasionally, for which its computational cost remains lower.

Since \(l\) is smooth in \(V_{\perp}\), problem (17) can be viewed as an unconstrained optimization problem. Optimization on the Stiefel manifold \(\mathbb{Q}\) requires preserving the orthogonality constraint in updating \(V_{\perp}\). The updating scheme involves retraction [17], which smoothly maps the tangent space \(T_{V_{\perp}}(\mathbb{Q}_{p \times d}) = \{ \Delta \in \mathbb{R}^{p \times d} : V_{\perp}^T \Delta + \Delta^T V_{\perp} = 0 \} \) onto the Stiefel manifold \(\mathbb{Q}_{p \times d}\) as is done in [17].

Let \(G\) denotes the Euclidean gradient of \(l\) with respect to \(V_{\perp}\), i.e., \(G_{ij} = (\partial l(V_{\perp})) / (\partial V_{\perp}(ij))\). For our interest, the Riemannian gradient of \(l\) with respect to \(V_{\perp}\), denoted by \(\nabla l\), is then given by

\[
\nabla l = WV_{\perp}. \tag{18}
\]

with \(W = GV_{\perp}^T - V_{\perp}G^T\) and \(G = (X - O)^T[(X - O)V_{\perp} - 1\mu^T - S] \).

Let \(V_{\perp}(r)\) be a function determining the new trial point with \(r\) as the step size. A valid updating scheme should guarantee that the new trial point lies on the manifold. This is ensured by using a Cayley transformation-based update as in [22]

\[
V_{\perp}(r) = (I + \frac{r}{2} W)^{-1}(I - \frac{r}{2} W)V_{\perp}. \tag{19}
\]

This curve always lies on the manifold \(\mathbb{Q}\) for any \(r\), and is also a descent curve passing the point \(V_{\perp}(0) = V_{\perp}\). A proper value for the step size \(r\) to guarantee convergence and efficiency in large-scale computation is chosen using BB stepsize update technique. Yet the inversion of the matrix inversion but still give the same solution as by (19). When \(d < p / 2\), we write \(W = A_1A_2^T\) with \(A_1 = [G V_{\perp}]\) and \(A_2 = [V_{\perp} - G]\), and apply the matrix inversion formula to get \(V_{\perp}(r) = V_{\perp} - rA_1(I + \tau A_2^T A_1)^{-1}A_2^TV_{\perp}\). This update formula involves the inversion of a \(2d \times 2d\) matrix, and turns out to be slightly faster than the original form. In the case of \(d \geq p / 2\), one possible idea is to approximate \(W\) by the product of two low-rank matrices. In the later sections, we also provide batch versions of the algorithm, which help divide its computational burden.

The entire stepwise procedure is outlined in Algorithm 3.

Algorithm 3: \(V_{\perp}\)-Opt ManiOptPackage

\[ k = 0. \]

\[ \tau_0 = 0.5. \]

repeat

\[ G = (X - O)^T[(X - O)V_{\perp} - 1\mu^T - S]. \]

\[ W = GV_{\perp}^T - V_{\perp}G^T. \]

\[ V_{\perp} = (I + \frac{\tau_k}{2} W)^{-1}(I - \frac{\tau_k}{2} W)V_{\perp}. \]

\[ \tau_{k+1} \text{ perform BB update.} \]

until \(\|l^{(k)} - l^{(k-1)}\|_\infty\) is small or \(k > k_{max}\)

The computations involved in the above manifold optimization algorithm, however, raise questions over its scalability to high-dimensional data. Also, the presence of too many free parameters (including those for the BB update) increases the model complexity as well as its sensitivity to fine-tuning.

Although the algorithm can minimize any smooth function over the Stiefel Manifold, the special structure of \(l(V_{\perp})\) in (17) calls for a rather simple yet effective method. If we substitute \((X - O)\) by \(Q\) and \((1\mu^T + S)\) by \(R\), it is interesting to note that the optimization looks similar to the orthogonal procurites rotation (PR) problem [23] as

\[
\min_{V_{\perp}} \frac{1}{2}\|QV_{\perp} - R\|_F^2, \quad \text{s.t. } V_{\perp}^T V_{\perp} = I. \tag{20}
\]

But the PR solution cannot be used directly, since \(V_{\perp} \in \mathbb{R}^{p \times d}\) is not a square orthonormal matrix. On the other hand, we can
plug in an iterative algorithm by alternatively performing a
linearized gradient descent and a PR solution to the nearest
orthonormal matrix estimation problem.

In terms of \( Q \) and \( R \), the Euclidean gradient of \( l \) over \( V \)
is \( Q^T (Q V - \otimes - R) \). Using the same linearization
method, the iterative estimation equation now looks like

\[
\tilde{V}^{(j+1)} = V^{(j)} - \frac{1}{\|Q\|^2} Q^T (Q V^{(j)} - R)
\]

(21)

where the step size is given by the inverse of the spectral
norm of the Hessian, which is \( \|Q\|^2 \).

The next step is to find the closest orthonormal matrix to
\( \tilde{V}^{(j+1)} \) so that the Stiefel manifold constraint is satisfied.
This can be mathematically written as

\[
\min_\mathcal{W} \| W - \tilde{V}^{(j+1)} \|_F, \quad \text{s.t. } W^T W = I.
\]

(22)

The Frobenius norm in the above can also be represented as

\[
\| W - \tilde{V}^{(j+1)} \|^2_F = \text{tr} \left[ (W - \tilde{V}^{(j+1)})^T (W - \tilde{V}^{(j+1)}) \right],
\]

(23)

Algorithm 4: \((V_{\perp})\)-Opt Lin+PR

Input: \( Q, R \)

Output: \( V \)

Initialize \( V_0 \in \mathbb{R}^{p \times d} \) as an orthonormal matrix.

\( j = 0, \Delta V^{(0)} = 0, \Delta V^{(j)} = 0 \).

repeat

\( j = j + 1, \Delta V^{(j)} = \frac{1}{\|Q\|^2} Q^T (Q V^{(j)} - R). \)

\( T = V^{(j)} - \Delta \tilde{A} V^{(j)} ] \).

Do SVD of \( T = U_w \Sigma_w V^T_w \).

\( U_{w}^{(w)} = \text{first } d \text{ columns of } U_w. \)

\( V^{(j)} = U_{w} U_{w}^{T} V_{w}^{T}. \)

until \( j > j_{\text{max}} \)

On decomposing the product inside the trace operator and
using \( W^T W = I \), the only term left that still depends on
\( W \) is \( \text{tr}[W^T \tilde{V}^{(j+1)}] \). Also, the negative sign in front of this
term changes (22) to a trace maximization problem. The new
optimization is given by

\[
\max_W \text{tr}[W^T \tilde{V}^{(j+1)}], \quad \text{s.t. } W^T W = I.
\]

(24)

The singular value decomposition (SVD) of \( \tilde{V}^{(j)} \) gives

\[
\tilde{V}^{(j+1)} = U_w \Sigma_w V_w^T \quad \Sigma_w = [\sigma_1 \ldots \sigma_d] (H^{(p-d) \times d}) \]

(25)

where \( \Sigma_w \) is the horizontal concatenation of a diagonal matrix of singular values and
an all-zero matrix. Also, \( U_w \) and \( V_w \) are orthogonal matrices in
\( \mathbb{R}^{p \times p} \) and \( \mathbb{R}^{d \times d} \), respectively. The SVD, in its economy size
version, can also be written as

\[
\tilde{V}^{(j+1)} = U_{\text{econ}} \Sigma_{\text{econ}} V_{w}^T \quad \Sigma_{\text{econ}} = \text{just a diagonal matrix with its diagonal entries}
\]

being \( \{\sigma_1 \ldots \sigma_d\} \) and \( U_{\text{econ}} \in \mathbb{R}^{p \times d} \) refers to the \( d \) columns
of \( U_w \) corresponding to the nonzero rows of \( \Sigma_w \).

We define an \( \mathbb{R}^{d \times d} \) orthogonal matrix as \( Z = V^T_w W^T U_{\text{econ}} \).
Now

\[
\text{tr}[W^T \tilde{V}^{(j+1)}] = \text{tr}[W^T U_{\text{econ}} \Sigma_{\text{econ}} V^T_w].
\]

(27)

The right-hand side in (27) can also be written as \( \text{tr}[Z \Sigma_{\text{w}}] \)
which can be further represented as

\[
\text{tr}[Z \Sigma_{\text{w}}] = \sum_{i=1}^{d} z_{ii} \sigma_i \leq \sum_{i=1}^{d} \sigma_i
\]

(28)

where the last inequality is based on von-Neumann’s trace
inequality [27] on the product of two square \( d \times d \) matrices
and the fact that the singular values of an orthogonal matrix \( Z \)
are all 1. Given that \( \sigma_i \) are all nonnegative, the upperbound
can be attained by letting \( Z \) to be equal to \( I_{d \times d} \). Thus, the
best approximation of \( W \) is given by \( U_{w}^{w} \Sigma_{\text{econ}} V^T_{w} \). Therefore, the
estimate of the OC space at the end of the next iteration is

\[
V_{\perp}^{(j+1)} = U_{w} \Sigma_{\text{econ}} V^T_w.
\]

(29)

The \( V_{\perp} \)-optimization problem may contain multiple local
minima of interest. In order to speed up the convergence, we make use of accelerated gradient methods like the momentum
technique which is popularly used in training neural
networks [13]. The update of \( V_{\perp} \) in the previous iteration is added to that of the current iteration after multiplying by a
factor \( \alpha \leq 1 \). This revised update is now used in Algorithm 4.
There are also other speedup techniques, such as Nesterov’s
accelerated gradient and AdaGrad [24].

We claim that the procedure followed in Algorithm 4 guarantees
that the subspace orthogonality constraint is satisfied and
the loss function is nonincreasing during the iteration:

\[
l(V^{(j+1)}_{\perp}) \leq l(V^{(j)}_{\perp}).
\]

In fact, this can be proved by defining a surrogate function

\[
g(\Xi, V_{\perp}) = l(V_{\perp}) + (\nabla l(V_{\perp}), \Xi - V_{\perp}) + \frac{\rho}{2} \| \Xi - V_{\perp} \|^2_F.
\]

(30)

Based on Taylor expansion

\[
g(\Xi, V_{\perp}) - l(\Xi) = \frac{\rho}{2} \| \Xi - V_{\perp} \|^2_F - (l(\Xi) - l(V_{\perp}) - (\nabla l(V_{\perp}), \Xi - V_{\perp})).
\]

(31)

Given \( V_{\perp} \), the problem of \( \min_{\Xi} \Xi - l g(\Xi, V_{\perp}) \) reduces to

\[
\min_{\Xi} \Xi - \left( V_{\perp} - \frac{1}{\rho} \nabla l(V_{\perp}) \right) \| F. \quad \text{s.t. } \Xi F = I.
\]

(32)

Therefore, it is easy to see that the inequality

\[
l(V_{\perp}^{(j+1)}) \leq g(V_{\perp}^{(j+1)}, V_{\perp}^{(j)}) \leq g(V_{\perp}^{(j)}, V_{\perp}^{(j)}) = l(V_{\perp}^{(j)})
\]

(33)

holds if \( \rho \geq \| Q \|^2 \).

C. Batch Version

Algorithm 4, with its computational burden, is not fast
enough to deal with usual high-dimensional \((p > n)\) real-
world data. For example, in case of camera images, the
dimensionality \( p \) is the number of pixels which can be of
the order of thousands or millions. The intrinsic dimensionality

It is only the projection space that matters and not the way the spanning vectors are calculated. In order to reach a local minima solution, the optimization can be approximated as solving $k$ subproblems such as

$$\min_{V_{\perp}} \frac{1}{2} \|QV_{\perp} - R_i\|^2_F, \text{ s.t. } V_{\perp}^T V_{\perp} = I$$

(35)

$\forall i = 1, 2, \ldots, k$. This considerably reduces the search space of each optimization and can help achieve a local minimum quickly. There is of course a tradeoff between $k$ and the accuracy of estimation. Algorithm 4 computes multiple SVDs of large matrices at a computational cost of $O(pd^2)$. Assuming the expected value of the $m_i$ values to be equal to $m$, the new computational complexity of the batch version gets reduced to $O(kpm^2)$, or also equal to $O(pdm)$. For high-dimensional data where $d$ is sufficiently high, this batch version plays a significant role in fastening the estimation process.

Algorithm 5: (Batch $V_{\perp}$)-Opt Version 1

Input: $X, O, \mu, S, [m_1, m_2, \ldots, m_k]$  
Output: $V_{\perp}$  

$j = 0$  
$Q = (X - O)$  
$R = I\mu^T + S$  
$s = 1$  
$SS = 0$  
$V_{\perp} = NULL$.

repeat

- Form $R_i$ by selecting $m_i$ columns of $R$ starting from the $s$th columns.  
- Call $V_{\perp s} = V_{\perp}$-opt($Q, R_i$).  
- $Z = (I - SS)V_{\perp s}$.  
- Concatenate $Z$ to the right of $V_{\perp}$.  
- $SS = SS + V_{\perp s}V_{\perp s}^T$.  
- $s = s + m_i$.  

until $i > k$

However, $V_{\perp}$ formed by concatenating these individual solutions for $V_{\perp j}$ may not generate an orthonormal set of bases for the OC subspace. Although $V_{\perp i}$ and $V_{\perp j}$ are separately orthonormal for distinct $i$ and $j$, they are not orthogonal to each other $V_{\perp i}^T V_{\perp j} \neq 0$. Applying the Grahm–Schmidt orthogonalization on the concatenated matrix $[V_{\perp 1} \ V_{\perp 2} \ \ldots \ V_{\perp \perp}]$ will let the computational complexity be $O(pd^2)$. However, we can take advantage of the special sequential structure of the individual $V_{\perp i}$'s. The first estimate $V_{\perp 1}$ is itself orthogonal and thus can be left untouched. Let $V_{\perp 1}^o = V_{\perp 1}$. The second estimate $V_{\perp 2}$ can be orthogonalized with respect to $V_{\perp 1}$ by

$$V_{\perp 2}^o = V_{\perp 2} - V_{\perp 1}(V_{\perp 1}^T V_{\perp 2})$$

(36)

which can also be written as

$$V_{\perp 2}^o = (I - V_{\perp 1}V_{\perp 1}^T)V_{\perp 2}$$

(37)

We can now horizontally concatenate $V_{\perp 1}^o$ and $V_{\perp 2}^o$ to form an orthonormal matrix with $m_1 + m_2$ columns. Similarly, $V_{\perp 3}$ can now be orthogonalized with respect to this concatenated matrix

$$V_{\perp 3}^o = (I - V_{\perp 1}^oV_{\perp 1}^{oT} + V_{\perp 2}^oV_{\perp 2}^{oT})V_{\perp 3}$$

(38)

However, $V_{\perp 1}^o V_{\perp 2}^{oT}$ has been precomputed. Thus, at each $ith$ iteration, we just have to compute $V_{\perp 2}^o V_{\perp 2}^{oT}$ and add it to previous sum $SS$ in the equation

$$V_{\perp i}^o = (I - (SS + V_{\perp 1}^{oT} + V_{\perp i}^{oT})V_{\perp i})V_{\perp i}$$

(39)

This considerably reduces the time of computing the orthonormal OC subspace and feeding it to $\mu$-$O$-$S$-opt alternatively. Algorithm 5 details the overall computation.

Algorithm 6: (Batch $V_{\perp}$)-Opt Version 2

Input: $X, m_i$  
Output: $V_{\perp}, O, \mu, S$  

$j = 0$  
$m_0 = 0$.

repeat

- $\sum_{i=1}^{j-1} m_i = s$.  
- Call $V_{\perp}$-opt on $X - \hat{O}^{(j)}$ to get $V_{\perp j} \in \mathbb{R}^{(p-s) \times m_j}$.  
- Call $(\mu, O, S)$-opt on $X$ but with dimension of $\mu$ and $S$ being $m_j$.  
- Find its orthogonal complement $V_{\perp j}^s$ which is of $p - s - m_j$-dimensions.  
- Deflate $X - \hat{O}^{(j+1)} = X - \hat{O}^{(j)} V_{\perp j}^s$.  
- Now $X - \hat{O}^{(j+1)} \in \mathbb{R}^{n \times (p-s-m_j)}$.  
- $j = j + 1$.

until $j \geq k$

Overall PC subspace $V = \bigcap V_{\perp}$.

And OC subspace estimation $V_{\perp} = \text{null}(VV^T)$.

Call $(\mu, O, S)$-opt one final time using $V_{\perp}$.

Another version of computing in batches is to have an iterative-estimation-and-deflation style [37] computing of the subspace vectors. This batch version of $\mathbb{R}^2$-PCP divides the work of the overall algorithm by alternatively calling $V_{\perp}$-opt and $(\mu, O, S)$-opt iteratively on the $p$-dimensional data $X$.  


only to estimate a partial subspace in each epoch. The final output is a $V_\perp$ that belongs to $\mathbb{R}^{p \times d}$. But it reduces the computational burden by dividing the $d$ into several $m_i$ values, such that $\sum_{i=1}^{d} m_i = d$. Each time we have an estimate for a partial rank-$m_i$ OC subspace, we deflate the design matrix accordingly. This ensures that the next set of estimated subspace vectors is independent of the ones estimated in the previous iteration. Though we have to run the $\mathbf{R}^2$-PCP $k$-times, each problem size is now much smaller and we can also have an early termination criterion for iterations, which can speed up the overall computation. Both the batch versions have been tested successfully in reducing the computational burden of the subspace estimation algorithm. It is up to the choice of the designer to decide between the two proposed techniques.

\section{V. Experiments}

\subsection{A. Experiments on Simulation Data}

In this section, we test the performance of $\mathbf{R}^2$-PCP against other robust subspace estimation methods on simulated data with e-Type $O$ and r-Type $S$ outliers. The techniques considered for comparison are RAPCA [29], ROBPCA [28], LRR [38], Go Decomposition (GoDec) [30], ROSL [32], and PCP [3] in its noise-considered stable version. In total, $n$ observations of $p$-dimensions are generated with only $r = 3$ being the dimension of the true PC subspace. Thus, only $r$ is relevant and enough in explaining the true data. The data varies along the three principal directions with $\Sigma = \text{diag}[64, 36, 16]$ and $\mu = \mathbf{0}$. It is now added with zero-mean Gaussian noise. For our experiments, we keep the noise variance as $\sigma^2 = 0.5$.

A percent variable $\psi$ is introduced to denote the percentage or sparsity of outlying entities. Since we focus here only on the e-Type $O$ and r-Type $S$ combination, $(\psi/2)$ fraction of the rows are outliers in the OC subspace, and additionally, $(\psi/2)$ fraction of the total elements in the $n \times p$ design matrix $X$ are large-valued garbage entries. Three different choices of $\psi \in \{4\%$, $10\%$, $16\%\}$ are experimented with and the magnitude of the outliers is taken to be large enough, say $L = 10$. This setting is used for creating a sparse matrix $S$ (with r-Type outliers) which has $n \times \psi$ of its rows as $[L, L, \ldots, L]$ and 0 otherwise. The sparse matrix $O$ (with e-Type outliers) has all 0 values except for the value of $L$ in $0.5 \times n \times p \times \psi$ of its randomly selected positions. Experiments are conducted for the low-dimensional case ($n < p$) with $n = 50$ and $p = 15$, and the high-dimensional case ($n > p$) with $n = 50$ and $p = 100$. The PC subspace estimation accuracy is measured in terms of cosine affinity (the cosine value of the largest canonical angle between the original PC subspace and the estimated PC subspace) and is averaged over 50 independent simulations and reported in Tables I–III.

A value of 1 indicates perfect estimation.

For quantifying the accuracy of outlier detection, three benchmark measures are used: the mean masking (M) and swamping (S) probabilities, and the rate of successful joint detection (JD). While masking probability is the fraction of undetected outliers, the swamping probability is the fraction of good points labeled as outliers. JD is the fraction of simulations with zero masking. An ideal method should have $M = 0$, $S = 0$, and $JD = 1$. In our case, masking is a much more serious problem than swamping, because the former can cause serious distortion in estimating true subspace. Thus, the upper bound parameters for outlier numbers are set to be $q_S = [0.75 \times n \times \psi]$ and $q_O = [0.75 \times n \times p \times \psi]$. Both correspond to approximately 1.5 times of the true number of outliers. Also, we ignore those e-Type outlier entities in matrix $O$ if the corresponding rows are detected as OC outliers (nonzero rows of $S$). The performance of $\mathbf{R}^2$-PCP, in terms of the above measures, is reported in Table IV.

As can be seen from the results, $\mathbf{R}^2$-PCP clearly edges over PCP and its other robust variants when it comes to how close the estimated subspace is to the true PC subspace in presence of both OS and OC outliers. The plain PCA, as expected, fails drastically. It is interesting to note that PCP performs as good as $\mathbf{R}^2$-PCP when there are very few outliers, like $\psi = 4\%$. However, when the percentage of outliers increases, the performance of other algorithms deteriorates drastically as compared with $\mathbf{R}^2$-PCP. Also, in the high-dimensional cases, our algorithm has an even bigger margin over the rest of the techniques. Only GoDec is as consistent but still its cosine affinity of estimated subspace is poorer than that of

\begin{table}[h]
\centering
\caption{Comparing Different Robust PCA Algorithms in Terms of Cosine Affinity for $n = 50$ and $p = 15$}
\begin{tabular}{|c|c|c|c|}
\hline
& $\psi = 4\%$ & $\psi = 10\%$ & $\psi = 16\%$ \\
\hline
R2PCP & 0.993 & 0.992 & 0.938 \\
PCA & 0.284 & 0.257 & 0.036 \\
PCP & 0.993 & 0.984 & 0.076 \\
RAPCA & 0.925 & 0.854 & 0.766 \\
ROBPCA & 0.945 & 0.865 & 0.921 \\
GoDec & 0.855 & 0.855 & 0.855 \\
ROSL & 0.893 & 0.494 & 0.081 \\
LRR & 0.992 & 0.945 & 0.058 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Comparing Different Robust PCA Algorithms in Terms of Cosine Affinity for $n = 50$ and $p = 100$}
\begin{tabular}{|c|c|c|c|}
\hline
& $\psi = 4\%$ & $\psi = 10\%$ & $\psi = 16\%$ \\
\hline
R2PCP & 0.966 & 0.944 & 0.911 \\
PCA & 0.043 & 0.085 & 0.035 \\
PCP & 0.939 & 0.098 & 0.047 \\
RAPCA & 0.483 & 0.241 & 0.113 \\
ROBPCA & 0.812 & 0.504 & 0.301 \\
GoDec & 0.732 & 0.772 & 0.772 \\
ROSL & 0.235 & 0.161 & 0.035 \\
LRR & 0.934 & 0.021 & 0.017 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Comparing Different Robust PCA Algorithms in Terms of Cosine Affinity for $n = 100$ and $p = 1000$}
\begin{tabular}{|c|c|c|c|}
\hline
& $\psi = 4\%$ & $\psi = 10\%$ & $\psi = 16\%$ \\
\hline
R2PCP & 0.862 & 0.802 & 0.787 \\
PCA & 0.007 & 0.003 & 0.007 \\
PCP & 0.009 & 0.006 & 0.006 \\
RAPCA & 0.175 & 0.130 & 0.057 \\
ROBPCA & 0.757 & 0.638 & 0.528 \\
GoDec & 0.363 & 0.363 & 0.363 \\
ROSL & 0.043 & 0.036 & 0.022 \\
LRR & 0.002 & 0.011 & 0.004 \\
\hline
\end{tabular}
\end{table}
TABLE IV
OUTLIER IDENTIFICATION RESULTS ON SIMULATION DATA WITH e-TYPE O AND r-TYPE S

<table>
<thead>
<tr>
<th>(n, p)</th>
<th>σ^2</th>
<th>ψ</th>
<th>M</th>
<th>S</th>
<th>JD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(50, 15)</td>
<td>0.5</td>
<td>10%</td>
<td>0.001</td>
<td>0.040</td>
<td>0.0975</td>
</tr>
<tr>
<td></td>
<td>16%</td>
<td>0.015</td>
<td>0.051</td>
<td>0.0966</td>
<td></td>
</tr>
<tr>
<td>(50, 100)</td>
<td>0.5</td>
<td>10%</td>
<td>0.000</td>
<td>0.038</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>16%</td>
<td>0.000</td>
<td>0.050</td>
<td>1.000</td>
<td></td>
</tr>
</tbody>
</table>

**R^2-PCP.** Table IV shows the outlier identification accuracy measures averaged over both S and O and it clearly shows that R^2-PCP is able to almost perfectly identify the locations of the outliers. The proposed batch versions were also tried out on the high-dimensional case with n = 100 and p = 1000. With no significant loss in overall subspace estimation accuracy, the time for each run of (V⊥)-opt could be reduced by almost half. The experiments were performed using MATLAB software on a single personal CPU.

**B. Face Recovery and Recognition**

This section deals with the real-world applications of R^2-PCP corresponding to the e-Type O and r-Type S combination. An important problem in the field of computer vision is learning the subspace of high-resolution images. According to [21], the set of images of a convex Lambertian object under various lighting conditions lies in a linear subspace spanning nine independent dimensions. Human faces can be considered to be partwise or approximately Lambertian. One of the primary applications of learning the true subspace can be the challenge of face recognition under varying environments. In reality, the acquired face images often suffer from cast shadows, specularities, saturation, and other kinds of aberrations. In this section, we use the R^2-PCP algorithm to remove such elementwise outliers in a set of face images of the same person mixed with some nonface random images which are now rowwise outliers, as shown in Fig. 5. The true underlying subspace is unknown. The experiment is performed on the Extended Yale Face Database B [25], [26], which includes aligned and cropped face images of 38 subjects (with the same frontal pose) under 64 different illumination conditions. We pick 31 face images of subject 22 and downsample these to 32 × 28 (=896) in order to save some computational cost.

In addition, we also generate four random (non-face) images of the same size, as shown in Fig. 5. Together, we get the 35 × 896 data matrix X, where we consider the first four observations (nonface images) as r-Type outliers in S, since they are totally independent of the factors that defines the real person’s face images. The rest 31 face images are assumed to be corrupted by an additive sparse outlier matrix in the OS (e-Type O), which refers to the cast shadows and specularities. In running the R^2-PCP algorithm, we set r = 9, q_s = 4, and q_s^0 = 0.04 × 35 × 896 = 1250, due to the sparsity assumption of the outliers and the insensitivity property of upper bound of the number of outliers.

Plotting eigenfaces is a way of visualizing the PC vectors or columns of the estimated subspace V by reordering them back into matrices of the size of original images. Fig. 6 shows eigenfaces formed by reshaping the vectors of the principal subspace estimated by PCP into the shape of original images in the top row. Similar eigenfaces generated by R^2-PCP are shown in the bottom row. It can be seen that a couple of the eigendimensions in the case of PCP look very noisy and may have absolutely nothing informative to describe the true subspace. In each row of Fig. 7, from left to right, we show in order the original face image, the detected outliers in the OS (white pixels represent nonzero entries in O) and the low-rank approximation B of the face images obtained by applying R^2-PCP. Clearly, the irregular pixels, like along the nose bridge, the forehead and the nose tip are detected as outliers. R^2-PCP can well compensate for these and provide good face recovery. Also, using our algorithm, we could perfectly identify the locations of the nonface images as OC outliers.

Face recognition is an important aspect of profile learning in online social network portals. A user may post pictures of scenic wallpapers, food, cartoons, beaches, or other kinds of images along with those containing his own face. Also, the face pictures may be marred by various elementwise OS space outliers. In this case, R^2-PCP is a one-shot method to identify the subspace by eliminating both kinds of outliers. About 280 training face images from the Yale Database belonging to 8 different persons, labeled 1 to 8, and some 20 real-world nonface images which are pictures of flowers, beaches, and cartoons. These outliers were also allotted a label randomly selected from 1 to 8. A sample set of the pictures has already been shown in Fig. 3. A 9-D subspace was estimated using PCA, robust PCA, and R^2-PCP and the data were projected onto this subspace as an act of feature transformation or dimensionality reduction. It is assumed that if the true subspace can be perfectly estimated, projection of the data points onto it can make the different classes more separable.

The test set consisted of 224 images of those 8 persons and the estimated projection space is also applied on them. The dimension reduced data were then fed to a linear classifier. Not only could R^2-PCP estimate a better subspace, but also detect...
the presence of outliers and subsequently remove them from subsequent learning. The rows in the projection of training data onto the subspace which correspond to the estimated rowwise outliers were removed while training the linear classifier in case of $R^2$-PCP. In Table V, it can be seen that the misclassification percentage in the case of $R^2$-PCP is considerably lower than PCA or PCP. This advantage can be attributed to its efficiency in learning a more accurate subspace along with removing unwanted rowwise and elementwise outliers reliably.

### C. Video Background Modeling

Finally, we also applied our algorithm in video background subtraction. A video sequence of a scene where the background is still and only few objects are mobile can be considered as an addition of a low-rank background video and an outlier video that contains just the moving objects. We used the popular hall video data set [18] for our experiments. However, we added another moving object, which can either be a bus, train, bird, or any obstruction in front of the video camera, that is either large or close enough to block the camera’s sight of the true background for at least a few frames. Thus in the current setup, the background information is completely lost in a few frames of the video. Such cases have a high probability of occurrence in real life situations. If a robust PCA algorithm fails to detect these frames as whole example outliers, the estimated background may not be the true fixed background and we may also have discrepancies in object tracking. In our experiments, this obstruction was a randomly...
generated pixel matrix of the size of each image frame in the video. Fig. 8 shows the low-rank ($B$) approximations of few frames of the video ordered chronologically from top to bottom obtained by PCP and $R^2$-PCP. The PCP still shows few instances of the moving people in its estimated background video as can be seen in the second column of Fig. 8. It totally loses track of the background in the frames when the cameras view is blocked temporarily. However, $R^2$-PCP performs much better in remaining truly robust in such cases. This is why the estimated background in the third column of Fig. 8 is consistent across all the frames shown. Also, $R^2$-PCP successfully captures all moving objects as outliers, whereas PCP estimates obscured frames by missing some of the corresponding foreground pixels. This can be specifically seen in the frames of the first row in Fig. 9. We also made a frame by frame comparison of the background estimated by our algorithm with some of the state-of-the-art low-rank subspace recovery-based video background estimation algorithms present in the LRS library [35]. The techniques considered for comparison are fast PCP, GoDec, Grassman average scalable robust PCA, Frank–Wolfe method-based robust PCA and mixture of Gaussians robust PCA. As can be seen in Fig. 10, our algorithm is pretty comparable with others or sometimes even better at the background estimation task.

![Fig. 9. PCP and $R^2$-PCP for tracking moving objects in frames arranged chronologically from top to bottom.](image)

**VI. Conclusion**

In this paper, we have proposed a novel design model for robust subspace learning that can deal with outliers both in the observation and OC space separately. Algorithms were developed to iteratively estimate the true subspace and outlying matrices. Experiments on artificially generated data clearly proved its efficiency over other robust PCA models. We were also able to compute a low-dimensional subspace of human face images and classification results on projected data indicate that the estimated subspace is more relevant to the learning task as compared to that extracted using PCA or PCP. $R^2$-PCP could identify the nonface images perfectly as $r$-type OC outliers. It was also successful in pointing out specular reflections and cast shadows as $e$-type OS outliers. Our model also finds semantic application in video surveillance specially when the given background is blocked from the camera’s view for certain amount of time. The recovery of the fixed background and moving objects in the foreground using $R^2$-PCP was shown to be much better in comparison to simple PCP. The reinforcement in our robust PCA analysis can be attributed to several factors like the usage of nonconvex sparsity-enforcing penalties, treating OC outliers with special care and accommodating whole sample rejection along with just elementwise anomalies. Although this efficiency is attained at a relatively higher cost of computation, we have outlined a few batch-iterative faster versions of the $V_{1,\perp}$-optimization. A detailed analysis of these batch algorithms on big data sets is yet to be done and is considered as one of our future works. There have been various modifications and improvements upon...
PCP in the literature and we believe that combining those changes with R²-PCP can lead to even powerful subspace learning methods. Such an approach can also be extended to nonlinear subspace representation algorithms like manifold learning and especially deep neural networks which have been lately very successful in various applications. It has been shown that generative deep neural networks [34] can flatten the highly entangled raw data manifolds and help capture the governing low-dimensional subspace. Considering the OC space outliers in such architectures can also provide interesting insights and results.

REFERENCES


Pratik Prabhanjan Brahama, photograph and biography not available at the time of publication.

Yiyuan She, photograph and biography not available at the time of publication.

Shijie Li, photograph and biography not available at the time of publication.

Jiadie Li, photograph and biography not available at the time of publication.

Dapeng Wu, photograph and biography not available at the time of publication.
AUTHOR QUERIES
AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your paper. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

AQ:1 = Please confirm whether the edits made in the financial information are OK.
AQ:2 = Please indicate the section numbers for the phrase “the later sections” throughout the article.
AQ:3 = Figs. 5–7 have been rearranged to ensure sequential order. Please confirm.
AQ:4 = Please provide the issue no. or month for ref. [12].
AQ:5 = Please provide the volume no., issue no., page range, and year for ref. [31].
AQ:6 = Please provide authors’ biographies.