1

Robust PCA by Controlling Sparsity in Model Residuals

Gonzalo Mateos
University of Rochester, USA

Georgios B. Giannakis
University of Minnesota, USA

1.1 Introduction

Principal component analysis (PCA) is the workhorse of high-dimensional data analysis and dimensionality reduction, with numerous applications in statistics, engineering, and the biobehavioral sciences; see, e.g., [Jol02]. Nowadays ubiquitous e-commerce sites, the Web, and urban traffic surveillance systems generate massive volumes of data. As a result, the problem of extracting the most informative, yet low-dimensional structure from high-dimensional datasets is of paramount importance [SGM14, HTF09]. To this end, PCA provides least-squares (LS) optimal linear approximants in $\mathbb{R}^q$ to a data set in ambient space $\mathbb{R}^p$, for $q \leq p$. The desired linear subspace is obtained from the q-dominant eigenvectors of the sample data covariance matrix, or equivalently from the q-dominant singular vectors of the data matrix [Jol02].

Data obeying postulated low-rank models include also outliers, which are samples not adhering to those nominal models. Unfortunately, LS is known to be very sensitive to outliers [RL87, HR09], and this undesirable property is inherited by PCA as well [Jol02]. Early efforts to robustify PCA have relied on robust estimates of the data covariance matrix; see, e.g., [Cam80]. Related approaches are driven from statistical physics [XY95], and also from M-estimators [dlTB03]. A fast algorithm for computer vision applications was put forth in [SRUB09]. Recently, polynomial-time algorithms with remarkable performance guarantees have emerged for low-rank matrix recovery in the presence of sparse – but otherwise

arbitrarily large – errors [CLMW11, CSPW11]. This pertains to an 'idealized robust' PCA setup, since those entries not affected by outliers are assumed error free. Stability in reconstructing the low-rank and sparse matrix components in the presence of 'dense' noise have been reported in [ZLW+10, XCS12]. A hierarchical Bayesian model was proposed to tackle the aforementioned low-rank plus sparse matrix decomposition problem in [DHC11].

In the present chapter, a robust PCA approach is pursued requiring minimal assumptions on the outlier model. A natural least-trimmed squares (LTS) PCA estimator is first shown closely related to an estimator obtained from an ℓ_0-(pseudo)norm-regularized criterion, adopted to fit a low-rank bilinear factor analysis model that explicitly incorporates an unknown *sparse* vector of outliers per datum (Section 1.2). As in compressive sampling [Tro06], efficient (approximate) solvers are obtained in Section 1.2.3, by surrogating the ℓ_0-norm of the outlier matrix with its closest convex approximant. This leads naturally to an M-type PCA estimator, which subsumes Huber's optimal choice as a special case [Fuc99]. Unlike Huber's formulation though, results here are not confined to an outlier contamination model. A tunable parameter controls the sparsity of the estimated matrix, and the number of outliers as a byproduct. Hence, effective data-driven methods to select this parameter are of paramount importance, and systematic approaches are pursued by efficiently exploring the entire *robustifaction* (a.k.a. homotopy) path of (group-) Lasso solutions [HTF09, YL06]. In this sense, the method here capitalizes on but *is not limited to* sparse settings where outliers are sporadic, since one can examine all sparsity levels along the robustification path. The outlier-aware generative data model and its sparsity-controlling estimator are quite general, since minor modifications discussed in [MG12, Sec. III-C] enable robustifiying linear regression [GMF+11], dictionary learning [TF10, MBPS10], and K-means clustering as well [HTF09, FKG11]. Section 1.3.2 deals with further modifications for bias reduction through nonconvex regularization, and automatic determination of the reduced dimension q is explored in Section 1.4 by drawing connections with nuclear-norm minimization [CLMW11, CSPW11].

Beyond its ties to robust statistics, the developed outlier-aware PCA framework is versatile to accommodate scalable *robust* algorithms to: i) track the low-rank signal subspace, as new data are acquired in real time (Section 1.4.1); and ii) determine principal components in (possibly) infinite-dimensional feature spaces, thus robustifying kernel PCA [SSM98], and spectral clustering as well [HTF09, p. 544] (Section 1.5). The vast literature on *non-robust* subspace tracking algorithms includes [Yan95, MBPS10], and [BNR10]; see also [HBS12] for a first-order algorithm that is robust to outliers and incomplete data. Relative to [HBS12], the online robust (OR)-PCA algorithm of [MMG15, MMG13b] (described in Section 1.4.1) is a second-order method, which minimizes an outlier-aware exponentially-weighted LS estimator of the low-rank factor analysis model. Since the outlier and subspace estimation tasks decouple nicely in OR-PCA, one can readily devise a first-order counterpart when minimal computational loads are at a premium. In terms of performance, online algorithms are known to be markedly faster than their batch alternatives [BNR10, HBS12], e.g., in the timely context of low-rank matrix completion [RFP10, RR13]. While the focus here is not on incomplete data records, extensions to account for missing data are immediate and have been reported in [MMG15].

Numerical tests with real data are presented throughout to corroborate the effectiveness of the proposed batch and online robust PCA schemes, when used to identify aberrant responses from a questionnaire designed to measure the Big-Five dimensions of personality traits [JNS08], as well as unveil communities in a (social) network of college football teams [GN02], and intruders from video surveillance data [dlTB03]. For additional comprehensive tests and comparisons with competing alternatives (omitted here due to lack of space), the reader is referred to [MG12, Sec. VII-A]. Concluding remarks are given in Section 1.6.

Notation: Bold uppercase (lowercase) letters will denote matrices (column vectors). Operators $(\cdot)'$ and $\mathrm{tr}(\cdot)$, will denote transposition and matrix trace, respectively. Vector $\mathrm{diag}(\mathbf{M})$ collects the diagonal entries of $\mathbf{M}$, whereas the diagonal matrix $\mathrm{diag}(\mathbf{v})$ has the entries of $\mathbf{v}$ on its diagonal. The ℓ_p-norm of $\mathbf{x} \in \mathbb{R}^n$ is $\|\mathbf{x}\|_p := \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ for $p \geq 1$; and $\|\mathbf{M}\|_F := \sqrt{\mathrm{tr}\left(\mathbf{M}\mathbf{M}'\right)}$ is the matrix Frobenious norm. The $n \times n$ identity matrix will be represented by $\mathbf{I}_n$, while $\mathbf{0}_n$ will denote the $n \times 1$ vector of all zeros, and $\mathbf{0}_{n\times m} := \mathbf{0}_n\mathbf{0}_m'$. Similar notation will be adopted for vectors (matrices) of all ones. The i-th vector of the canonical basis in $\mathbb{R}^n$ will be denoted by $\mathbf{b}_{n,i}$, $i = 1, \ldots, n$.

1.2 Robustifying PCA

Consider the standard PCA formulation, in which a set of training data $\mathcal{T}_y := \{\mathbf{y}_n\}_{n=1}^N$ in the p-dimensional Euclidean *input* space is given, and the goal is to find the best q-rank ($q \leq p$) linear approximation to the data in $\mathcal{T}_y$; see e.g., [Jol02]. Unless otherwise stated, it is assumed throughout that the value of q is given. One approach to solving this problem, is to adopt a low-rank bilinear (factor analysis) model

$$\mathbf{y}_n = \mathbf{m} + \mathbf{U}\mathbf{s}_n + \mathbf{e}_n, \quad n = 1, \ldots, N \tag{1.1}$$

where $\mathbf{m} \in \mathbb{R}^p$ is a location (mean) vector; matrix $\mathbf{U} \in \mathbb{R}^{p\times q}$ has orthonormal columns spanning the signal subspace; $\{\mathbf{s}_n\}_{n=1}^N$ are the so-termed *principal components*, and $\{\mathbf{e}_n\}_{n=1}^N$ are zero-mean i.i.d. random errors. The unknown variables in (1.1) can be collected in $\mathcal{V} := \{\mathbf{m}, \mathbf{U}, \{\mathbf{s}_n\}_{n=1}^N\}$, and they are estimated using the LS criterion as

$$\min_{\mathcal{V}} \sum_{n=1}^N \|\mathbf{y}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n\|_2^2, \quad \text{s. to} \quad \mathbf{U}'\mathbf{U} = \mathbf{I}_q. \tag{1.2}$$

PCA in (1.2) is a nonconvex optimization problem due to the bilinear terms $\mathbf{U}\mathbf{s}_n$, yet a global optimum $\hat{\mathcal{V}}$ can be shown to exist; see e.g., [Yan95]. The resulting estimates are $\hat{\mathbf{m}} = \sum_{n=1}^N \mathbf{y}_n/N$ and $\hat{\mathbf{s}}_n = \hat{\mathbf{U}}'(\mathbf{y}_n - \hat{\mathbf{m}})$, $n = 1, \ldots, N$; while $\hat{\mathbf{U}}$ is formed with columns equal to the q-dominant right singular vectors of the $N \times p$ data matrix $\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_N]'$ [HTF09, p. 535]. The principal components (entries of) $\mathbf{s}_n$ are the projections of the centered data points $\{\mathbf{y}_n - \hat{\mathbf{m}}\}_{n=1}^N$ onto the signal subspace. Equivalently, PCA can be formulated based on maximum variance, or, minimum reconstruction error criteria; see e.g., [Jol02].

1.2.1 Least-Trimmed Squares PCA

Given training data $\mathcal{T}_x := \{\mathbf{x}_n\}_{n=1}^N$ possibly contaminated with outliers, the goal here is to develop a robust estimator of $\mathcal{V}$ that requires minimal assumptions on the outlier model. Note that there is an explicit notational differentiation between: i) the data in $\mathcal{T}_y$ which adhere to the nominal model (1.1); and ii) the given data in $\mathcal{T}_x$ that may also contain outliers, i.e., those $\mathbf{x}_n$ not adhering to (1.1). Building on LTS regression [RL87], the desired robust estimate $\hat{\mathcal{V}}_{LTS} := \{\hat{\mathbf{m}}, \hat{\mathbf{U}}, \{\hat{\mathbf{s}}_n\}_{n=1}^N\}$ for a prescribed $\nu < N$ can be obtained via the following LTS PCA estimator [cf. (1.2)]

$$\hat{\mathcal{V}}_{LTS} := \arg\min_{\mathcal{V}} \sum_{n=1}^{\nu} r_{[n]}^2(\mathcal{V}), \quad \text{s. to} \quad \mathbf{U}'\mathbf{U} = \mathbf{I}_q \tag{1.3}$$

where $r_{[n]}^2(\mathcal{V})$ is the n-th order statistic among the squared residual norms $r_1^2(\mathcal{V}), \ldots, r_N^2(\mathcal{V})$, and $r_n(\mathcal{V}) := \|\mathbf{x}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n\|_2$. The so-termed *coverage* ν determines the breakdown point

of the LTS PCA estimator [RL87], since the $N-\nu$ largest residuals are absent from the estimation criterion in (1.3). Beyond this universal outlier-rejection property, the LTS-based estimation offers an attractive alternative to robust linear regression due to its high breakdown point and desirable analytical properties, namely $\sqrt{N}$-consistency and asymptotic normality under mild assumptions [RL87].

Because (1.3) is a nonconvex optimization problem, a nontrivial issue pertains to the existence of the proposed LTS PCA estimator, i.e., whether or not (1.3) attains a minimum. Fortunately, existence of $\hat{\mathcal{V}}_{LTS}$ can be readily established as follows: i) for each subset of $\mathcal{T}$ with cardinality ν (there are $\binom{N}{\nu}$ such subsets), solve the corresponding PCA problem to obtain a unique candidate estimator per subset; and ii) pick $\hat{\mathcal{V}}_{LTS}$ as the one among all $\binom{N}{\nu}$ candidates with the minimum cost. Albeit conceptually simple, the aforementioned solution procedure is combinatorially complex, and thus intractable except for small sample sizes N. Algorithms to obtain approximate LTS solutions in large-scale linear regression problems are available; see e.g., [RL87].

REMARK 1.1 In most PCA formulations data in $\mathcal{T}_y$ are typically assumed zero mean. This is without loss of generality, since nonzero-mean training data can always be rendered zero mean, by subtracting the sample mean $\sum_{n=1}^{N}\mathbf{y}_n/N$ from each $\mathbf{y}_n$. In modeling zero-mean data, the known vector $\mathbf{m}$ in (1.1) can obviously be neglected. When outliers are present however, data in $\mathcal{T}_x$ are not necessarily zero mean, and it is unwise to center them using the non-robust sample mean estimator which has a breakdown point equal to zero [RL87]. Towards robustifying PCA, a more sensible approach is to estimate $\mathbf{m}$ robustly, and jointly with $\mathbf{U}$ and the principal components $\{\mathbf{s}_n\}_{n=1}^{N}$. For this reason $\mathbf{m}$ is kept as a variable in $\mathcal{V}$ and estimated via (1.3).

1.2.2 Robust Statistics Meets Sparse Recovery

Instead of discarding large residuals, the alternative approach here explicitly accounts for outliers in the low-rank data model (1.1). This becomes possible through the vector variables $\{\mathbf{o}_n\}_{n=1}^{N}$ one per training datum $\mathbf{x}_n$, which take the value $\mathbf{o}_n\neq\mathbf{0}_p$ whenever datum n is an outlier, and $\mathbf{o}_n=\mathbf{0}_p$ otherwise. Thus, the outlier-aware factor analysis model is

$$\mathbf{x}_n=\mathbf{y}_n+\mathbf{o}_n=\mathbf{m}+\mathbf{U}\mathbf{s}_n+\mathbf{e}_n+\mathbf{o}_n,\qquad n=1,\ldots,N \tag{1.4}$$

where $\mathbf{o}_n$ can be deterministic or random with unspecified distribution. In the *underdetermined* linear system of equations (1.4), both $\mathcal{V}$ as well as the $N\times p$ matrix $\mathbf{O}:=[\mathbf{o}_1,\ldots,\mathbf{o}_N]'$ are unknown. The percentage of outliers dictates the degree of *sparsity* (number of zero rows) in $\mathbf{O}$. Sparsity control will prove instrumental in efficiently estimating $\mathbf{O}$, rejecting outliers as a byproduct, and consequently arriving at a *robust* estimator of $\mathcal{V}$. To this end, a natural criterion for controlling outlier sparsity is to seek the estimator [cf. (1.2)]

$$\{\hat{\mathcal{V}},\hat{\mathbf{O}}\}=\arg\min_{\mathcal{V},\mathbf{O}}\|\mathbf{X}-\mathbf{1}_N\mathbf{m}'-\mathbf{S}\mathbf{U}'-\mathbf{O}\|_F^2+\lambda_0\|\mathbf{O}\|_0,\quad\text{s. to }\mathbf{U}'\mathbf{U}=\mathbf{I}_q \tag{1.5}$$

where $\mathbf{X}:=[\mathbf{x}_1,\ldots,\mathbf{x}_N]'\in\mathbb{R}^{N\times p}$, $\mathbf{S}:=[\mathbf{s}_1,\ldots,\mathbf{s}_N]'\in\mathbb{R}^{N\times q}$, and $\|\mathbf{O}\|_0$ denotes the nonconvex ℓ_0-norm that is equal to the number of nonzero rows of $\mathbf{O}$. Vector (group) sparsity in the rows $\hat{\mathbf{o}}_n$ of $\hat{\mathbf{O}}$ can be directly controlled by tuning the parameter $\lambda_0\geq 0$.

As with compressive sampling and sparse modeling schemes that rely on the ℓ_0-norm [Tro06], the robust PCA problem (1.5) is NP-hard [Nat95]. In addition, the sparsity-controlling estimator (1.5) is intimately related to LTS PCA, as asserted in the following proposition. (A detailed proof is also included since it is instructive towards revealing the link between both estimators.)

PROPOSITION 1.1 If $\{\hat{\mathcal{V}}, \hat{\mathbf{O}}\}$ minimizes (1.5) with λ_0 chosen such that $\|\hat{\mathbf{O}}\|_0 = N - \nu$, then $\hat{\mathcal{V}}_{LTS} = \hat{\mathcal{V}}$.

PROOF 1.1 Given λ_0 such that $\|\hat{\mathbf{O}}\|_0 = N - \nu$, the goal is to characterize $\hat{\mathcal{V}}$ as well as the positions and values of the nonzero rows of $\hat{\mathbf{O}}$. Because $\|\hat{\mathbf{O}}\|_0 = N - \nu$, the last term in the cost of (1.5) is constant, hence inconsequential to the minimization. Upon defining $\hat{\mathbf{r}}_n := \mathbf{x}_n - \hat{\mathbf{m}} - \hat{\mathbf{U}}\hat{\mathbf{s}}_n$, the rows of $\hat{\mathbf{O}}$ satisfy

$$\hat{\mathbf{o}}_n = \begin{cases} \mathbf{0}_p, & \|\hat{\mathbf{r}}_n\|_2 \leq \sqrt{\lambda_0} \\ \hat{\mathbf{r}}_n, & \|\hat{\mathbf{r}}_n\|_2 > \sqrt{\lambda_0} \end{cases}, \quad n = 1, \ldots, N. \tag{1.6}$$

This follows by noting first that (1.5) is separable across the rows of $\mathbf{O}$. For each $n = 1, \ldots, N$, if $\hat{\mathbf{o}}_n = \mathbf{0}_p$ then the optimal cost becomes $\|\hat{\mathbf{r}}_n - \hat{\mathbf{o}}_n\|_2^2 + \lambda_0\|\hat{\mathbf{o}}_n\|_0 = \|\hat{\mathbf{r}}_n\|_2^2$. If on the other hand $\hat{\mathbf{o}}_n \neq \mathbf{0}_p$, the optimality condition for $\mathbf{o}_n$ yields $\hat{\mathbf{o}}_n = \hat{\mathbf{r}}_n$, and thus the cost reduces to λ_0. In conclusion, for the chosen value of λ_0 it holds that $N - \nu$ squared residuals effectively do not contribute to the cost in (1.5).

To determine $\hat{\mathcal{V}}$ and the row support of $\hat{\mathbf{O}}$, one alternative is to exhaustively test all $\binom{N}{N-\nu} = \binom{N}{\nu}$ admissible row-support combinations. For each one of these combinations (indexed by j), let $\mathcal{S}_j \subset \{1, \ldots, N\}$ be the index set describing the row support of $\hat{\mathbf{O}}^{(j)}$, i.e., $\hat{\mathbf{o}}_n^{(j)} \neq \mathbf{0}_p$ if and only if $n \in \mathcal{S}_j$; and $|\mathcal{S}_j| = N - \nu$. By virtue of (1.6), the corresponding candidate $\hat{\mathcal{V}}^{(j)}$ solves $\min_{\mathcal{V}} \sum_{n \in \mathcal{S}_j} r_n^2(\mathcal{V})$ subject to $\mathbf{U}'\mathbf{U} = \mathbf{I}_q$, while $\hat{\mathcal{V}}$ is the one among all $\{\hat{\mathcal{V}}^{(j)}\}$ that yields the least cost. Recognizing the aforementioned solution procedure as the one for LTS PCA outlined in Section 1.2.1, it follows that $\hat{\mathcal{V}}_{LTS} = \hat{\mathcal{V}}$. ■

The importance of Proposition 1.1 is threefold. First, it formally justifies model (1.4) and its estimator (1.5) for robust PCA, in light of the well documented merits of LTS [RL87]. Second, it establishes a connection between the seemingly unrelated fields of robust statistics and sparsity-aware estimation. Third, problem (1.5) lends itself naturally to efficient (approximate) solvers based on convex relaxation, the subject dealt with next.

1.2.3 Sparsity-Controlling Outlier Rejection

Recall that the row-wise ℓ_2-norm sum $\|\mathbf{B}\|_{2,r} := \sum_{n=1}^{N} \|\mathbf{b}_n\|_2$ of matrix $\mathbf{B} := [\mathbf{b}_1, \ldots, \mathbf{b}_N]' \in \mathbb{R}^{N \times p}$ is the closest convex approximation of $\|\mathbf{B}\|_0$ [Tro06]. This property motivates relaxing problem (1.5) to

$$\min_{\mathcal{V}, \mathbf{O}} \|\mathbf{X} - \mathbf{1}_N\mathbf{m}' - \mathbf{S}\mathbf{U}' - \mathbf{O}\|_F^2 + \lambda_2\|\mathbf{O}\|_{2,r}, \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q. \tag{1.7}$$

The nondifferentiable ℓ_2-norm regularization term encourages row-wise (vector) sparsity on the estimator of $\mathbf{O}$, a property that has been exploited in diverse problems in engineering, statistics, and machine learning [HTF09]. A noteworthy representative is the group Lasso [YL06], a popular tool for joint estimation and selection of grouped variables in linear regression.

REMARK 1.2 In computer vision applications for instance where robust PCA schemes are particularly attractive, one may not wish to discard the entire (vectorized) images $\mathbf{x}_n$, but only specific pixels deemed as outliers [dlTB03]. This can be accomplished by replacing $\|\mathbf{O}\|_{2,r}$ in (1.7) with $\|\mathbf{O}\|_1 := \sum_{n=1}^{N} \|\mathbf{o}_n\|_1$, a Lasso-type regularization that encourages entry-wise sparsity in $\hat{\mathbf{O}}$.

After the relaxation it is pertinent to ponder on whether problem (1.7) still has the potential of providing robust estimates $\hat{\mathcal{V}}$ in the presence of outliers. The answer is positive, since (1.7) is equivalent to an M-type PCA estimator

$$\min_{\mathcal{V}} \sum_{n=1}^{N} \rho_v(\mathbf{x}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n), \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q \tag{1.8}$$

where $\rho_v : \mathbb{R}^p \to \mathbb{R}$ is a vector extension to Huber's convex loss function [HR09]; namely

$$\rho_v(\mathbf{r}) := \begin{cases} \|\mathbf{r}\|_2^2, & \|\mathbf{r}\|_2 \leq \lambda_2/2 \\ \lambda_2\|\mathbf{r}\|_2 - \lambda_2^2/4, & \|\mathbf{r}\|_2 > \lambda_2/2 \end{cases} . \tag{1.9}$$

For a detailed proof of the equivalence, see [MG12].

M-type estimators (including Huber's) adopt a fortiori an ϵ-contaminated probability distribution for the outliers, and rely on minimizing the *asymptotic* variance of the resultant estimator for the least favorable distribution of the ϵ-contaminated class (asymptotic min-max approach) [HR09]. The assumed degree of contamination specifies the tuning parameter λ_2 in (1.9) (and thus the threshold for deciding the outliers in M-estimators). In contrast, the present approach is universal in the sense that it is not confined to any assumed class of outlier distributions, and can afford a data-driven selection of the tuning parameter. In a nutshell, optimal M-estimators can be viewed as a special case of the present formulation only for a specific choice of λ_2, which is not obtained via a data-driven approach, but from distributional assumptions instead.

All in all, the sparsity-controlling role of the tuning parameter $\lambda_2 \geq 0$ in (1.7) is central, since model (1.4) and the equivalence of (1.7) with (1.8) suggest that λ_2 is a robustness-controlling constant. Data-driven approaches to select λ_2 are described in detail under Section 1.3.1. Before delving into algorithmic issues to solve (1.7), a remark is in order.

REMARK 1.3 The recent upsurge of research toward compressive sampling and parsimonious signal representations hinges on signals being sparse, either naturally, or, after projecting them on a proper basis. Here instead, a neat link is established between sparsity and a fundamental aspect of statistical inference, namely that of robustness against outliers. It is argued that key to robust methods is the control of sparsity in *model residuals*, i.e., those entries in matrix $\mathbf{O}$, even when the signals in $\mathcal{V}$ are not (necessarily) sparse.

1.3 Algorithms and Real Data Tests

To optimize (1.7) iteratively for a given value of λ_2, an alternating minimization (AM) algorithm is adopted which cyclically updates $\mathbf{m}(k) \to \mathbf{S}(k) \to \mathbf{U}(k) \to \mathbf{O}(k)$ per iteration $k = 1, 2, \ldots$. AM algorithms are also known as block-coordinate-descent methods in the optimization parlance; see e.g., [Ber99, Tse01]. To update each of the variable groups, (1.7) is minimized while fixing the rest of the variables to their most up-to-date values. While the overall problem (1.7) is not jointly convex with respect to (w.r.t.) $\{\mathbf{S}, \mathbf{U}, \mathbf{O}, \mathbf{m}\}$, fixing all but one of the variable groups yields subproblems that are efficiently solved, and attain a unique solution.

Towards deriving the updates at iteration k and arriving at the desired algorithm, note first that the mean update is $\mathbf{m}(k) = (\mathbf{X} - \mathbf{O}(k))'\mathbf{1}_N/N$. Next, form the centered and outlier-compensated data matrix $\mathbf{X}_o(k) := \mathbf{X} - \mathbf{1}_N\mathbf{m}(k)' - \mathbf{O}(k-1)$. The principal components are readily given by

$$\mathbf{S}(k) = \arg\min_{\mathbf{S}} \|\mathbf{X}_o(k) - \mathbf{S}\mathbf{U}(k-1)'\|_F^2 = \mathbf{X}_o(k)\mathbf{U}(k-1).$$

Continuing the cycle, $\mathbf{U}(k)$ solves

$$\min_{\mathbf{U}} \|\mathbf{X}_o(k) - \mathbf{S}(k)\mathbf{U}'\|_F^2, \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q$$

a constrained LS problem also known as reduced-rank *Procrustes rotation* [ZHT06]. The minimizer is given in analytical form in terms of the left and right singular vectors of $\mathbf{X}_o'(k)\mathbf{S}(k)$ [ZHT06, Thm. 4]. In detail, one computes the SVD of $\mathbf{X}_o'(k)\mathbf{S}(k) = \mathbf{L}(k)\mathbf{D}(k)\mathbf{R}'(k)$ and updates $\mathbf{U}(k) = \mathbf{L}(k)\mathbf{R}'(k)$. Next, the minimization of (1.7) w.r.t. $\mathbf{O}$ is an orthonormal group Lasso problem. As such, it decouples across rows $\mathbf{o}_n$ giving rise to N ℓ_2-norm regularized subproblems, namely

$$\mathbf{o}_n(k) = \arg\min_{\mathbf{o}} \|\mathbf{r}_n(k) - \mathbf{o}\|_2^2 + \lambda_2\|\mathbf{o}\|_2, \quad n = 1, \ldots, N$$

where $\mathbf{r}_n(k) := \mathbf{x}_n - \mathbf{m}(k) - \mathbf{U}(k)\mathbf{s}_n(k)$. The respective solutions are given by (see e.g., [PWH11])

$$\mathbf{o}_n(k) = \frac{\mathbf{r}_n(k)(\|\mathbf{r}_n(k)\|_2 - \lambda_2/2)_+}{\|\mathbf{r}_n(k)\|_2}, \quad n = 1, \ldots, N \tag{1.10}$$

where $(\cdot)_+ := \max(\cdot, 0)$. For notational convenience, these N parallel vector soft-thresholded updates are denoted as $\mathbf{O}(k) = \mathcal{S}\left[\mathbf{X} - \mathbf{1}_N\mathbf{m}'(k-1) - \mathbf{S}(k)\mathbf{U}'(k), (\lambda_2/2)\mathbf{I}_N\right]$ under Algorithm 1, where the thresholding operator $\mathcal{S}$ sets the entire outlier vector $\mathbf{o}_n(k)$ to zero whenever $\|\mathbf{r}_n(k)\|_2$ does not exceed $\lambda_2/2$, in par with the group sparsifying property of group Lasso. Interestingly, this is the same rule used to decide if datum $\mathbf{x}_n$ is deemed an outlier, in the equivalent formulation (1.8) which involves Huber's loss function. Whenever an ℓ_1-norm regularizer is adopted as discussed in Remark 1.2, the only difference is that updates (1.10) boil down to soft-thresholding the scalar entries of $\mathbf{r}_n(k)$.

The entire AM solver is tabulated under Algorithm 1, indicating also the recommended initialization. Algorithm 1 is conceptually interesting, since it explicitly reveals the intertwining between the outlier identification process, and the PCA low-rank model fitting based on the outlier compensated data $\mathbf{X}_o(k)$. The AM solver is also computationally efficient. Computing the $N \times q$ matrix $\mathbf{S}(k) = \mathbf{X}_o(k)\mathbf{U}(k-1)$ requires Npq operations per iteration, and equally costly is to obtain $\mathbf{X}_o'(k)\mathbf{S}(k) \in \mathbb{R}^{p\times q}$. The cost of computing the SVD of $\mathbf{X}_o'(k)\mathbf{S}(k)$ is of order $\mathcal{O}(pq^2)$, while the rest of the operations including the row-wise soft-thresholdings to yield $\mathbf{O}(k)$ are linear in both N and p. In summary, the total cost of Algorithm 1 is roughly $k_{\max}\mathcal{O}(Np + pq^2)$, where $k_{\max}$ is the number of iterations required for convergence (typically $k_{\max} = 5$ to 10 iterations suffice). Because $q \leq p$ is typically small, Algorithm 1 is attractive computationally both under the classic setting where $N > p$, and p is not large; as well as in high-dimensional data settings where $p \gg N$, a situation typically arising e.g., in microarray data analysis.

Because each of the optimization problems in the per-iteration cycles has a unique minimizer, and the nondifferentiable regularization only affects one of the variable groups ($\mathbf{O}$), the general results of [Tse01] apply to establish convergence of Algorithm 1. Specifically, as $k \to \infty$ the iterates generated by Algorithm 1 converge to a stationary point of (1.7).

1.3.1 Selection of λ_2: Robustification Paths

Selecting λ_2 controls the number of outliers rejected. But this choice is challenging because existing techniques such as cross-validation are not effective when outliers are present [RL87]. To this end, systematic data-driven approaches were devised in [GMF+11], which e.g., require a rough estimate of the percentage of outliers, or, robust estimates $\hat{\sigma}_e^2$ of the nominal noise variance that can be obtained using median absolute deviation (MAD) schemes [HR09]. These approaches can be adapted to the robust PCA setting considered

Algorithm 1 : Batch robust PCA solver

Set $\mathbf{U}(0) = \mathbf{I}_p(:, 1:q)$ and $\mathbf{O}(0) = \mathbf{0}_{N\times p}$.
for $k = 1, 2, \ldots$ **do**
  Update $\mathbf{m}(k) = (\mathbf{X} - \mathbf{O}(k-1))'\mathbf{1}_N/N$.
  Form $\mathbf{X}_o(k) = \mathbf{X} - \mathbf{1}_N\mathbf{m}'(k) - \mathbf{O}(k-1)$.
  Update $\mathbf{S}(k) = \mathbf{X}_o(k)\mathbf{U}(k-1)$.
  Obtain $\mathbf{L}(k)\mathbf{D}(k)\mathbf{R}(k)' = \text{svd}[\mathbf{X}_o'(k)\mathbf{S}(k)]$ and update $\mathbf{U}(k) = \mathbf{L}(k)\mathbf{R}'(k)$.
  Update $\mathbf{O}(k) = \mathcal{S}\left[\mathbf{X} - \mathbf{1}_N\mathbf{m}'(k) - \mathbf{S}(k)\mathbf{U}'(k), (\lambda_2/2)\mathbf{I}_N\right]$.
end for

here, and leverage the *robustification paths* of (group-)Lasso solutions [cf. (1.7)], which are defined as the solution paths corresponding to $\|\hat{\mathbf{o}}_n\|_2$, $n = 1, \ldots, N$, for all values of λ_2. As λ_2 decreases, more vectors $\hat{\mathbf{o}}_n$ enter the model signifying that more of the training data are deemed to contain outliers.

Consider then a grid of G_λ values of λ_2 in the interval $[\lambda_{\min}, \lambda_{\max}]$, evenly spaced on a logarithmic scale. Typically, $\lambda_{\max}$ is chosen as the minimum λ_2 value such that $\hat{\mathbf{O}} \neq \mathbf{0}_{N\times p}$, while $\lambda_{\min} = \epsilon\lambda_{\max}$ with $\epsilon = 10^{-4}$, say. Because Algorithm 1 converges quite fast, (1.7) can be efficiently solved over the grid of G_λ values for λ_2. In the order of hundreds of grid points can be easily handled by initializing each instance of Algorithm 1 (per value of λ_2) using *warm starts* [HTF09]. This means that multiple instances of (1.7) are solved for a sequence of decreasing λ_2 values, and the initialization of Algorithm 1 per grid point corresponds to the solution obtained for the immediately preceding value of λ_2 in the grid. For sufficiently close values of λ_2, one expects that the respective solutions will also be close (the row support of $\hat{\mathbf{O}}$ will most likely not change), and hence Algorithm 1 will converge after few iterations.

Based on the G_λ samples of the robustification paths and the prior knowledge available on the outlier model (1.4), a couple of alternatives described next are possible for selecting the 'best' value of λ_2 in the grid. A comprehensive survey of options can be found in [GMF+11].

Number of outliers is known: By direct inspection of the robustification paths one can determine the range of values for λ_2, such that the number of nonzero rows in $\hat{\mathbf{O}}$ equals the known number of outliers sought. Zooming-in to the interval of interest, and after discarding the identified outliers, K-fold cross-validation methods can be applied to determine the 'best' λ_2^*.

Nominal noise covariance matrix is known: Given $\mathbf{\Sigma}_e := E[\mathbf{e}_n\mathbf{e}_n']$, one can proceed as follows. Consider the estimates $\hat{\mathcal{V}}_g$ obtained using (1.7) after sampling the robustification path for each point $\{\lambda_{2,g}\}_{g=1}^G$. Next, pre-whiten those residuals corresponding to training data not deemed as containing outliers; i.e., form $\hat{\mathcal{R}}_g := \{\bar{\mathbf{r}}_{n,g} = \mathbf{\Sigma}_e^{-1/2}(\mathbf{x}_n - \hat{\mathbf{m}}_g - \hat{\mathbf{U}}_g\hat{\mathbf{s}}_{n,g}) : n \text{ s. to } \hat{\mathbf{o}}_n = \mathbf{0}\}$, and find the sample covariance matrices $\{\hat{\mathbf{\Sigma}}_{\bar{r},g}\}_{g=1}^G$. The winner $\lambda_2^* := \lambda_{2,g^*}$ corresponds to the grid point minimizing an absolute variance deviation criterion, namely $g^* := \arg\min_g |\text{tr}[\hat{\mathbf{\Sigma}}_{\bar{r},g}] - p|$.

1.3.2 Bias reduction through nonconvex regularization

Instead of substituting $\|\mathbf{O}\|_0$ in (1.5) by its closest convex approximation, namely $\|\mathbf{O}\|_{2,r}$, letting the surrogate function to be nonconvex can yield tighter approximations, and improve the statistical properties of the estimator. In rank minimization problems for instance, the logarithm of the determinant of the unknown matrix has been proposed as a smooth

surrogate to the rank [FHB03]; an alternative to the convex nuclear norm in e.g., [RFP10]. Nonconvex penalties such as the smoothly clipped absolute deviation (SCAD) have been also adopted to reduce bias [FL01], present in uniformly weighted ℓ_1-norm regularized estimators such as (1.7) [HTF09, p. 92]. In the context of sparse signal reconstruction, the ℓ_0-norm of a vector was surrogated in [CWB08] by the logarithm of the geometric mean of its elements; see also [RLS09].

Building on this last idea, consider approximating (1.5) by the formulation

$$\min_{\mathcal{V},\mathbf{O}} \|\mathbf{X}-\mathbf{1}_N\mathbf{m}'-\mathbf{S}\mathbf{U}'-\mathbf{O}\|_F^2+\lambda_0\sum_{n=1}^{N}\log(\|\mathbf{o}_n\|_2+\delta), \quad \text{s. to } \mathbf{U}'\mathbf{U}=\mathbf{I}_q \tag{1.11}$$

where the small positive constant δ is introduced to avoid numerical instability. Since the surrogate term in (1.11) is concave, the overall minimization problem is nonconvex and admittedly more complex to solve than (1.7). Local methods based on iterative linearization of $\log(\|\mathbf{o}_n\|_2+\delta)$ around the current iterate $\mathbf{o}_n(k)$, are adopted to minimize (1.11). Skipping details that can be found in [KG11], application of the majorization-minimization technique to (1.11) leads to an iteratively-reweighted version of (1.7), whereby $\lambda_2 \leftarrow \lambda_0 w_n(k)$ is used for updating $\mathbf{o}_n(k)$ in Algorithm 1. Specifically, per $k=1,2,\ldots$ one updates

$$\mathbf{O}(k)=\mathcal{S}\left[\mathbf{X}-\mathbf{1}_N\mathbf{m}'(k-1)-\mathbf{S}(k)\mathbf{U}'(k),(\lambda_0/2)\text{diag}(w_1(k),\ldots,w_N(k))\right]$$

where the weights are given by $w_n(k)=\left(\|\mathbf{o}_n(k-1)\|_2+\delta\right)^{-1}$, $n=1,\ldots,N$. Note that the thresholds vary both across rows (indexed by n), and across iterations. If the value of $\|\mathbf{o}_n(k-1)\|_2$ is small, then in the next iteration the regularization term $\lambda_0 w_n(k)\|\mathbf{o}_n\|_2$ has a large weight, thus promoting shrinkage of that entire row vector to zero. If $\|\mathbf{o}_n(k-1)\|_2$ is large, the cost in the next iteration downweighs the regularization, and places more importance to the LS component of the fit.

All in all, the idea is to start from the solution of (1.7) for the 'best' λ_2, which is obtained using Algorithm 1. This initial estimate is refined after runnning a few iterations of the iteratively-reweighted counterpart to Algorithm 1. Extensive numerical tests suggest that even a couple iterations of this second stage refinement suffices to yield improved estimates $\hat{\mathcal{V}}$, in comparison to those obtained from (1.7); see also the detailed numerical tests in [MG12] . The improvements can be leveraged to bias reduction – and its positive effect with regards to outlier support estimation – also achieved by similar *weighted* norm regularizers proposed for linear regression [HTF09, p. 92].

1.3.3 Video surveillance

To validate the proposed approach to robust PCA, Algorithm 1 was tested to perform background modeling from a sequence of video frames; an approach that has found widespread applicability for intrusion detection in video surveillance systems. The experiments were carried out using the dataset studied in [dlTB03], which consists of $N=520$ images ($p=120\times 160$) acquired from a static camera during two days. The illumination changes considerably over the two day span, while approximately 40% of the training images contain people in various locations. For $q=10$, both standard PCA and the robust PCA (Algorithm 1) were applied to build a low-rank background model of the scenery captured by the camera. For robust PCA, ℓ_1-norm regularization on $\mathbf{O}$ was adopted to identify outliers at a pixel level. The outlier sparsity-controlling parameter was chosen as $\lambda_2=9.69\times 10^{-4}$, whereas a single iteration of the reweighted scheme in Section 1.3.2 was run to reduce the bias in $\hat{\mathbf{O}}$.

Results are shown in Fig. 1.1, for three representative images. The first column comprises the original frames from the training set, while the second column shows the corresponding

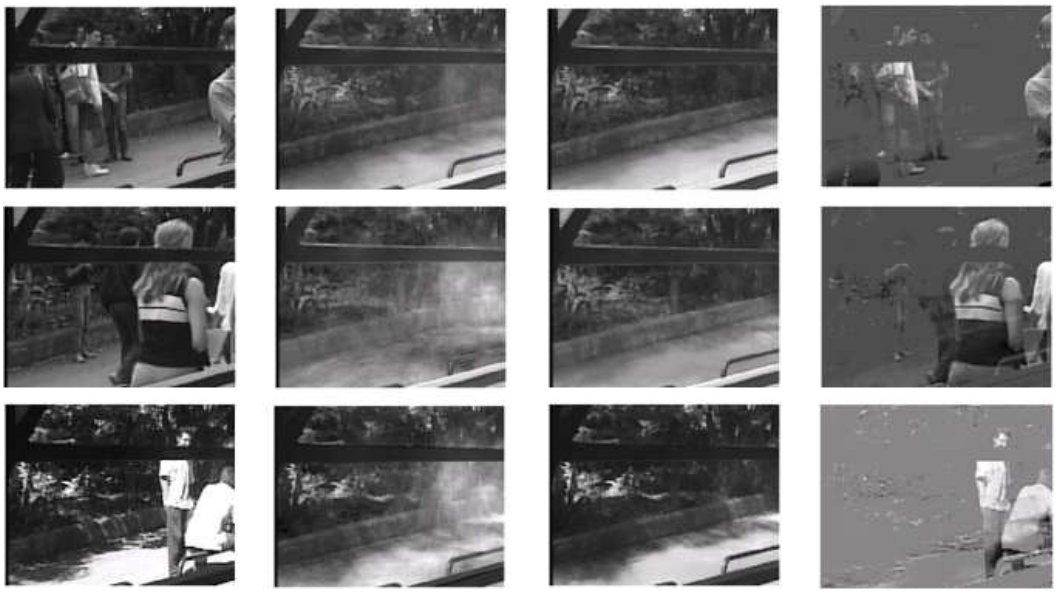

FIGURE 1.1 Background modeling for video surveillance. First column: original frames. Second column: PCA reconstructions, where the presence of undesirable 'ghostly' artifacts is apparent. Third column: robust PCA reconstructions, which recover the illumination changes while successfully subtracting the people. Fourth column: outliers in $\hat{\mathbf{o}}$, which mostly capture the people and abrupt changes in illumination.

PCA image reconstructions. The presence of undesirable 'ghostly' artifacts is apparent, since PCA is unable to completely separate the people from the background. The third column illustrates the robust PCA reconstructions, which recover the illumination changes while successfully subtracting the people. The fourth column shows the reshaped outlier vectors $\hat{\mathbf{o}}_n$, which mostly capture the people and abrupt changes in illumination. See also [MG12] for additional comparisons with competing methods, including e.g., the algorithm in [dlTB03].

1.3.4 Robust measurement of the Big Five personality factors

The 'Big Five' are five factors ($q = 5$) of personality traits, namely extraversion, agreeableness, conscientiousness, neuroticism, and openness; see e.g., [JNS08]. The Big Five inventory (BFI) on the other hand, is a brief questionnaire (44 items in total) tailored to measure the Big Five dimensions. Subjects taking the questionnaire are asked to rate in a scale from 1 (disagree strongly) to 5 (agree strongly), items of the form 'I see myself as someone who is talkative'. Each item consists of a short phrase correlating (positively or negatively) with one factor; see e.g., [JNS08, pp. 157-58] for a copy of the BFI and scoring instructions.

Robust PCA is used to identify aberrant responses from real BFI data comprising the Eugene-Springfield community sample [Gol08]. The rows of $\mathbf{X}$ contain the $p = 44$ item responses for each one of the $N = 437$ subjects under study. For $q = 5$ and $\lambda_2 = 5.6107$ corresponding to $\|\hat{\mathbf{O}}\|_0 = 100$, Fig. 1.2 depicts the norm of the 40 largest outliers. There is an unmistakable break in the scree plot and the 8 largest values are declared as outliers by robust PCA. As a means of validating these results, the following procedure is adopted. Based on the BFI scoring key [JNS08], a list of all pairs of items hypothesized to yield positively correlated responses is formed. For each n, one counts the 'inconsistencies' defined as the number of times that subject n's ratings for these pairs differ in more than four, in

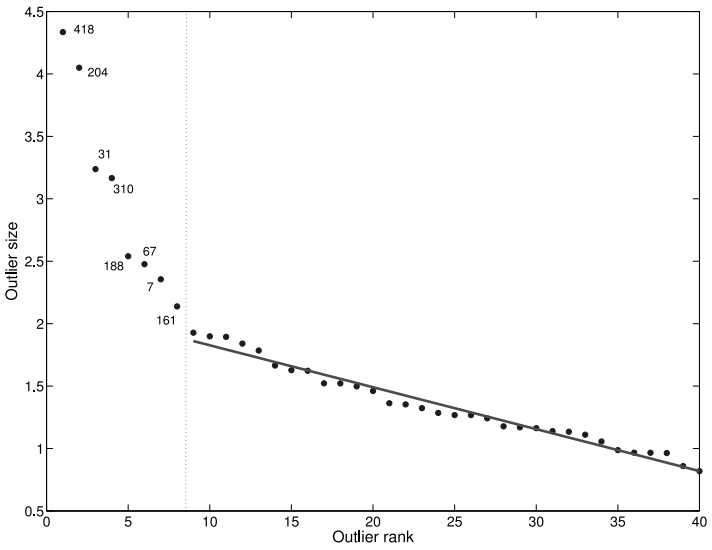

FIGURE 1.2 Pseudo scree plot of outlier size $(\|\hat{\mathbf{o}}_n\|_2)$; the 40 largest outliers are shown. Robust PCA declares the largest 8 as aberrant responses.

absolute value. Interestingly, after rank-ordering all subjects in terms of this inconsistency score, one finds that $n = 418$ ranks highest with a count of 17, $n = 204$ ranks second (10), and overall the eight outliers found rank in the top twenty.

1.4 Connections with Nuclear-Norm Minimization

Recall that $q \leq p$ is the dimensionality of the subspace where the outlier-free data (1.1) are assumed to live in, or equivalently, $q = \text{rank}[\mathbf{Y}]$ in the absence of noise. So far, q was assumed known and fixed. This is reasonable in e.g., compression/quantization, where a target distortion-rate tradeoff dictates the maximum q. In other cases, the physics of the problem may render q known. This is indeed the case in array processing for direction-of-arrival estimation, where q is the dimensionality of the so-termed *signal subspace*, and is given by the number of plane waves impinging on a uniform linear array; see e.g., [Yan95].

Other applications however, call for signal processing tools that can determine the 'best' q, as well as robustly estimate the underlying low-dimensional subspace $\mathbf{U}$ from data $\mathbf{X}$. Noteworthy representatives for this last kind of problems include unveiling traffic volume anomalies in large-scale networks [MMG13b, MMG13a], and automatic intrusion detection from video surveillance frames [dlTB03, CLMW11], just to name a few. A related approach in this context is (stable) principal components pursuit (PCP) [ZLW$^+$10, XCS12], which solves

$$\min_{\mathbf{L},\mathbf{O}} \|\mathbf{X} - \mathbf{L} - \mathbf{O}\|_F^2 + \lambda_* \|\mathbf{L}\|_* + \lambda_2 \|\mathbf{O}\|_{2,r} \tag{1.12}$$

with the objective of reconstructing the low-rank matrix $\mathbf{L} \in \mathbb{R}^{N \times p}$, as well as the sparse

matrix of outliers $\mathbf{O}$ in the presence of dense noise with known variance.* Note that $\|\mathbf{L}\|_*$ denotes the matrix nuclear norm, a convex surrogate to rank[$\mathbf{L}$] defined as the sum of the singular values of $\mathbf{L}$. The same way that the ℓ_2-norm regularization promotes sparsity in the rows of $\hat{\mathbf{O}}$, the nuclear norm encourages a low-rank $\hat{\mathbf{L}}$ since it effects sparsity in the vector of singular values of $\mathbf{L}$. Upon solving the convex optimization problem (1.12), it is possible to obtain $\hat{\mathbf{L}} = \hat{\mathbf{S}}\hat{\mathbf{U}}'$ using the SVD. Interestingly, (1.12) does not fix (or require the knowledge of) rank[$\mathbf{L}$] a fortiori, but controls it through the tuning parameter λ_*. Adopting a Bayesian framework, a similar problem was considered in [DHC11].

Instead of assuming that q is known, suppose that only an upper bound $\bar{q}$ is given. Then, the class of feasible noise-free low-rank matrix components of $\mathbf{Y}$ in (1.1) admit a factorization $\mathbf{L} = \mathbf{S}\mathbf{U}'$, where $\mathbf{S}$ and $\mathbf{U}$ are $N \times \bar{q}$ and $p \times \bar{q}$ matrices, respectively. Building on the ideas used in the context of finding minimum rank solutions of linear matrix equations [RFP10], an alternative approach to robustifying PCA is to solve [cf. (1.7)]

$$\min_{\mathbf{U},\mathbf{S},\mathbf{O}} \|\mathbf{X} - \mathbf{S}\mathbf{U}' - \mathbf{O}\|_F^2 + \frac{\lambda_*}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{S}\|_F^2) + \lambda_2\|\mathbf{O}\|_{2,r}. \tag{1.13}$$

Different from (1.12) and (1.7), a Frobenius-norm regularization on both $\mathbf{U}$ and $\mathbf{S}$ is adopted to control the dimensionality of the estimated subspace $\hat{\mathbf{U}}$. Relative to (1.7), $\mathbf{U}$ in (1.13) is not constrained to be orthonormal. It is certainly possible to include the mean vector $\mathbf{m}$ in the cost of (1.13), as well as an ℓ_1-norm regularization for entrywise outliers. The main motivation behind choosing the Frobenius-norm regularization comes from the equivalence of (1.12) with (1.13) provided rank[$\hat{\mathbf{L}}$] $\leq \bar{q}$, which follows by adapting the results in [RFP10, Lemma 5.1] to the problem formulation considered here; see also the seminal work in [SRJ04, SS05].

Even though problem (1.13) is nonconvex, the number of optimization variables is reduced from $2Np$ to $Np + (N + p)\bar{q}$, which becomes significant when $\bar{q}$ is small and both N and p are large. Also note that the dominant Np-term in the variable count of (1.13) is due to $\mathbf{O}$, which is sparse and can be efficiently handled. While the factorization $\mathbf{L} = \mathbf{S}\mathbf{U}'$ could have also been introduced in (1.12) to reduce the number of unknowns, the cost in (1.13) is separable and much simpler to optimize using e.g., an AM solver comprising the iterations tabulated in [MG12, Alg. 2]; see also the discussion on subspace trackers in the ensuing section.

Because (1.13) is a nonconvex optimization problem, most solvers one can think of will at most provide convergence guarantees to a stationary point that may not be globally optimum. Interestingly, the ensuing proposition adapted from [MMG13a, Prop. 1] and [BM05] offers a certificate for stationary points of (1.13), qualifying them as global optima of (1.12).

PROPOSITION 1.2 If $\{\bar{\mathbf{U}}, \bar{\mathbf{S}}, \bar{\mathbf{O}}\}$ is a stationary point of (1.13) and $\|\mathbf{X} - \bar{\mathbf{S}}\bar{\mathbf{U}}' - \bar{\mathbf{O}}\|_2 \leq \lambda_*/2$, then $\{\hat{\mathbf{L}} := \bar{\mathbf{S}}\bar{\mathbf{U}}', \hat{\mathbf{O}} := \bar{\mathbf{O}}\}$ is the optimal solution of (1.12).

The usefulness of the separable Frobenius-norm regularization in (1.13) is further illustrated next, in the context of robust subspace tracking.

*Actually, [ZLW+10] considers entrywise outliers and adopts an ℓ_1-norm regularization on $\mathbf{O}$.

1.4.1 Robust Subspace Tracking

E-commerce and Internet-based retailing sites, the World Wide Web, and video surveillance systems generate huge volumes of data, which far outweigh the ability of personal computers to analyze them in real time. Furthermore, observations are oftentimes acquired *sequentially in time*, which motivates updating previously obtained 'analytics' rather than re-computing new ones from scratch each time a new datum becomes available [SKMG14]. This calls for low-complexity real-time (adaptive) algorithms for robust subspace tracking; see e.g., [MMG15].

One possible adaptive counterpart to (1.13) is the exponentially-weighted LS (EWLS) estimator found by [MMG13b]

$$\min_{\{\mathcal{V},\mathbf{O}\}} \sum_{n=1}^{N} \beta^{N-n} \left[\|\mathbf{x}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n - \mathbf{o}_n\|_2^2 + \frac{\lambda_*}{2\sum_{u=1}^{N}\beta^{N-u}} \|\mathbf{U}\|_F^2 + \frac{\lambda_*}{2}\|\mathbf{s}_n\|_2^2 + \lambda_2\|\mathbf{o}_n\|_2 \right] \tag{1.14}$$

where $\beta \in (0,1]$ is a forgetting factor. In this context, n should be understood as a temporal variable, indexing the instants of data acquisition. Note that in forming the EWLS estimator (1.14) at time N, the entire history of data $\{\mathbf{x}_n\}_{n=1}^N$ is incorporated in the real-time estimation process. Whenever $\beta < 1$, past data are exponentially discarded thus enabling operation in nonstationary environments. For the infinite memory case ($\beta = 1$) on the other hand, the formulation (1.14) coincides with the batch estimator (1.13). This is the reason for the time-varying weight normalizing $\|\mathbf{U}\|_F^2$.

A provably convergent subspace tracker is developed in [MMG13b], based on AM of (1.14). In a nutshell, each time a new datum is acquired, outlier estimates are formed via the Lasso [HTF09, p. 68], and the low-rank subspace is refined using recursive LS. For situations were reducing computational complexity is critical, an online stochastic gradient algorithm based on Nesterov's acceleration technique is developed as well [MMG13b]. In a stationary setting, the asymptotic subspace estimates obtained offer the well-documented performance guarantees of the batch stable PCP estimator [cf. (1.12) and Proposition 1.2].

Subspace tracking has a long history in signal processing. An early noteworthy representative is the projection approximation subspace tracking (PAST) algorithm [Yan95]; see also [YK88]. Recently, an algorithm (termed GROUSE) for tracking subspaces from incomplete observations was put forth in [BNR10], based on incremental gradient descent iterations on the Grassmannian manifold of subspaces. Recent analysis has shown that GROUSE can converge locally at an expected linear rate [BW13], and that it is tightly related to the incremental SVD algorithm [Bal13]. PETRELS is a second-order recursive least-squares (RLS)-type algorithm, that extends the seminal PAST iterations to handle missing data [CEC13]. As noted in [DMK11], the performance of GROUSE is limited by the existence of barriers in the search path on the Grassmanian, which may lead to GROUSE iterations being trapped at local minima; see also [CEC13]. Lack of regularization in PETRELS can also lead to unstable (even divergent) behaviors, especially when the amount of missing data is large. Accordingly, the convergence results for PETRELS are confined to the full-data setting where the algorithm boils down to PAST [CEC13]. When outliers are present, robust counterparts can be found in [QV11, HBS12, QVLH14].

REMARK 1.4 Towards addressing the scalability issue outlined at the beginning of this section, the decomposability of the Frobenius-norm regularizer in (1.13) has also been recently exploited for parallel processing across multiple processors when solving large-scale matrix completion problems [RR13], or to unveil network anomalies [MMG13a]. Specifically, [MMG13a] puts forth a general framework for *decentralized* sparsity-regularized rank minimization adopting the alternating-direction method of multipliers [BPC+10].

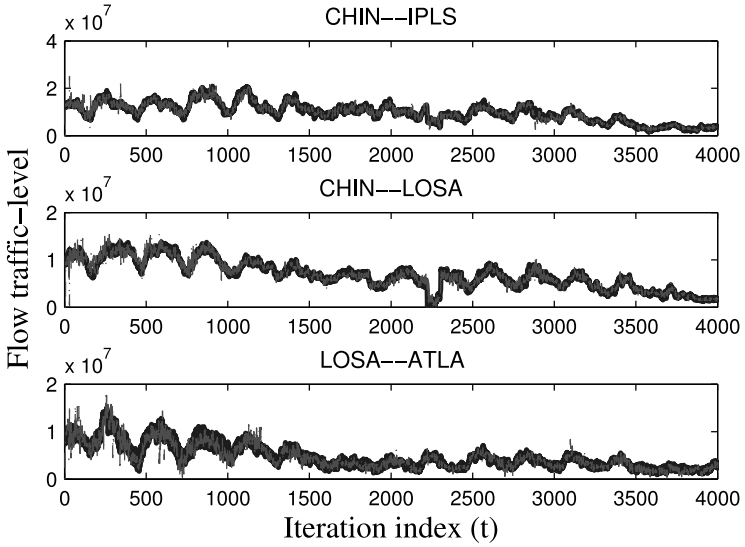

FIGURE 1.3 Online estimated (dashed gray) versus true (solid black) OD flow traffic for 75% missing data, and three representative flows measured from the operation of Internet2.

1.4.2 Tracking Internet Traffic Flows

Accurate estimation of origin-to-destination (OD) flow traffic in the backbone of large-scale Internet Protocol (IP) networks is of paramount importance for proactive network security and management tasks [Kol09]. Several experimental studies have demonstrated that OD flow traffic exhibits a low-intrinsic dimensionality, mainly due to common temporal patterns across OD flows, and periodic trends across time [LPC+04]. However, due to the massive number of OD pairs and the high volume of traffic, measuring the traffic of all possible OD flows is impossible for all practical purposes [LPC+04, Kol09]. Only the traffic level for a small fraction of OD flows can be measured via the NetFlow protocol [LPC+04].

Here, aggregate OD-flow traffic is collected from the operation of the Internet2 network (Internet backbone across USA) during December 8-28, 2003 containing 121 OD pairs. The measured OD flows contain spikes (anomalies or outliers), yielding the data stream $\{\mathbf{x}_n\} \in \mathbb{R}^{121}$. The detailed description of the considered dataset can be found in [MMG13b]. When only 25% of the total OD flows are sampled by Netflow, Fig. 1.3 depicts how the OR-PCA algorithm in [MMG15] accurately tracks three representative OD flows.

1.5 Robustifying Kernel PCA

Kernel (K)PCA is a generalization to (linear) PCA, seeking principal components in a *feature space* nonlinearly related to the *input space* where the data in $\mathcal{T}_x$ live [SSM98]. KPCA has been shown effective in performing nonlinear feature extraction for pattern recognition [SSM98]. In addition, connections between KPCA and spectral clustering [HTF09, p. 548] motivate well the KPCA method developed in this section, to robustly identify cohesive subgroups (communities) from social network data.

Consider a nonlinear function $\boldsymbol{\phi} : \mathbb{R}^p \to \mathcal{H}$, that maps elements from the input space $\mathbb{R}^p$ to a feature space $\mathcal{H}$ of arbitrarily large – possibly infinite – dimensionality. Given transformed data $\mathcal{T}_{\mathcal{H}} := \{\boldsymbol{\phi}(\mathbf{x}_n)\}_{n=1}^N$, the proposed approach to robust KPCA fits the

Algorithm 2 : Robust KPCA solver

Initialize $\mathbf{\Omega}(0) = \mathbf{0}_{N\times N}$, $\mathbf{S}(0)$ randomly, and form $\mathbf{K} = \mathbf{\Phi}'\mathbf{\Phi}$.
for $k = 1, 2, \ldots$ **do**
  Update $\boldsymbol{\mu}(k) = [\mathbf{I}_n - \mathbf{\Omega}(k-1)]\mathbf{1}_N/N$.
  Form $\mathbf{\Phi}_o(k) = \mathbf{I}_N - \boldsymbol{\mu}(k)\mathbf{1}_N' - \mathbf{\Omega}(k-1)$.
  Update $\mathbf{\Upsilon}(k) = \mathbf{\Phi}_o(k)\mathbf{S}(k-1)[\mathbf{S}'(k-1)\mathbf{S}(k-1) + (\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1}$.
  Update $\mathbf{S}(k) = \mathbf{\Phi}_o'(k)\mathbf{K}\mathbf{\Upsilon}(k)[\mathbf{\Upsilon}(k)'\mathbf{K}\mathbf{\Upsilon}(k) + (\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1}$.
  Form $\boldsymbol{\rho}_n(k) = \mathbf{b}_{N,n} - \boldsymbol{\mu}(k) - \mathbf{\Upsilon}(\boldsymbol{k})\mathbf{s}_n(k)$, $n = 1, \ldots, N$.
  Form $\mathbf{\Lambda}(k) = \text{diag}\left(\frac{(\boldsymbol{\rho}_1'(k)\mathbf{K}\boldsymbol{\rho}_1(k) - \frac{\lambda_2}{2})_+}{\boldsymbol{\rho}_1'(k)\mathbf{K}\boldsymbol{\rho}_1(k)}, \ldots, \frac{(\boldsymbol{\rho}_N'(k)\mathbf{K}\boldsymbol{\rho}_N(k) - \frac{\lambda_2}{2})_+}{\boldsymbol{\rho}_N'(k)\mathbf{K}\boldsymbol{\rho}_N(k)}\right)$.
  Update $\mathbf{\Omega}(k) = [\mathbf{I}_N - \boldsymbol{\mu}(k)\mathbf{1}_N' - \mathbf{\Upsilon}(\boldsymbol{k})\mathbf{S}'(k)]\mathbf{\Lambda}(k)$.
end for

model

$$\boldsymbol{\phi}(\mathbf{x}_n) = \mathbf{m} + \mathbf{U}\mathbf{s}_n + \mathbf{e}_n + \mathbf{o}_n, \quad n = 1, \ldots, N \tag{1.15}$$

by solving ($\mathbf{\Phi} := [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_N)]$)

$$\min_{\mathbf{U},\mathbf{S},\mathbf{O}} \|\mathbf{\Phi}' - \mathbf{1}_N\mathbf{m}' - \mathbf{S}\mathbf{U}' - \mathbf{O}\|_F^2 + \frac{\lambda_*}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{S}\|_F^2) + \lambda_2\|\mathbf{O}\|_{2,r}. \tag{1.16}$$

It is certainly possible to adopt the criterion (1.7) as well, but (1.16) is chosen here for simplicity in exposition. Except for the principal components' matrix $\mathbf{S} \in \mathbb{R}^{N\times\bar{q}}$, both the data and the unknowns in (1.16) are now vectors/matrices of generally infinite dimension. In principle, this challenges the optimization task since it is impossible to store, or, perform updates of such quantities directly.

Interestingly, this hurdle can be overcome by endowing $\mathcal{H}$ with the structure of a reproducing kernel Hilbert space (RKHS), where inner products between any two members of $\mathcal{H}$ boil down to evaluations of the reproducing kernel $K_{\mathcal{H}} : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, i.e., $\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle_{\mathcal{H}} = K_{\mathcal{H}}(\mathbf{x}_i, \mathbf{x}_j)$. Specifically, it is possible to form the kernel matrix $\mathbf{K} := \mathbf{\Phi}'\mathbf{\Phi} \in \mathbb{R}^{N\times N}$, without directly working with the vectors in $\mathcal{H}$. This so-termed *kernel trick* is the crux of most kernel methods in machine learning [HTF09], including kernel PCA [SSM98]. The problem of selecting $K_{\mathcal{H}}$ (and $\boldsymbol{\phi}$ indirectly) will not be considered here.

Building on these ideas, it is asserted next that Algorithm 1 can be *kernelized*, to solve (1.16) at affordable computational complexity and memory storage requirements that do not depend on the dimensionality of $\mathcal{H}$. A proof of Proposition 1.3 is available in [MG12].

PROPOSITION 1.3 For $k \geq 1$, the sequence of iterates generated by Algorithm 1 when applied to solve (1.16) can be written as $\mathbf{m}(k) = \mathbf{\Phi}\boldsymbol{\mu}(k)$, $\mathbf{U}(k) = \mathbf{\Phi}\mathbf{\Upsilon}(k)$, and $\mathbf{O}'(k) = \mathbf{\Phi}\mathbf{\Omega}(k)$. The quantities $\boldsymbol{\mu}(k) \in \mathbb{R}^N$, $\mathbf{\Upsilon}(k) \in \mathbb{R}^{N\times\bar{q}}$, and $\mathbf{\Omega}(k) \in \mathbb{R}^{N\times N}$ are recursively updated as in Algorithm 2, without the need of operating with vectors in $\mathcal{H}$.

Proposition 1.3 asserts that if the iterates are initialized with outlier estimates in the range space of $\mathbf{\Phi}$, then all subsequent iterates will admit a similar expansion in terms of feature vectors. This is weaker than claiming that each minimizer of (1.16) admits such an expansion – the latter would require checking whether the regularization term in (1.16) satisfies the conditions of the Representer Theorem [SHS01].

In order to run the robust KPCA algorithm (tabulated as Algorithm 2), one does not have to store or process the quantities $\mathbf{m}(k)$, $\mathbf{U}(k)$, and $\mathbf{O}(k)$. As per Proposition 1.3, the iterations of a provably convergent AM solver can be equivalently carried out by cycling through *finite-dimensional* 'sufficient statistics' $\boldsymbol{\mu}(k) \to \mathbf{\Upsilon}(k) \to \mathbf{S}(k) \to \mathbf{\Omega}(k)$. In other

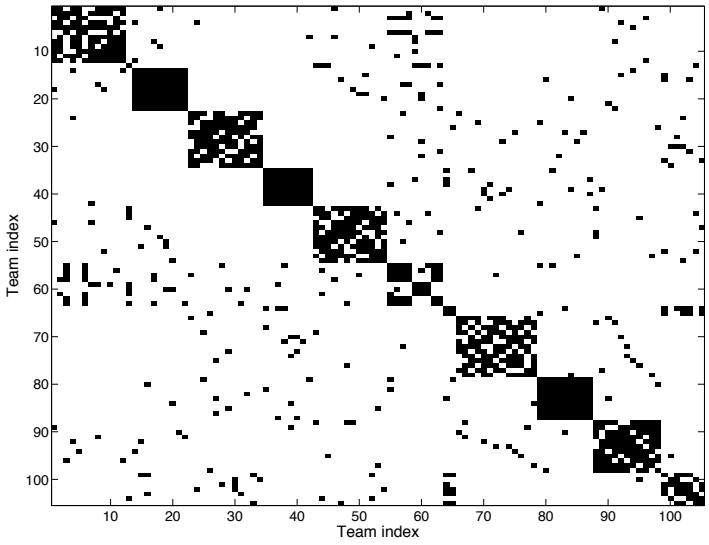

FIGURE 1.4 Entries of $\mathbf{K}$ after removing the outliers, where rows and columns are permuted to reveal the clustering structure found by robust KPCA. The eleven-conference (community) structure is apparent.

words, the iterations of the robust kernel PCA algorithm are devoid of algebraic operations among vectors in $\mathcal{H}$. Recall that the size of matrix $\mathbf{S}$ is independent of the dimensionality of $\mathcal{H}$.

Because $\mathbf{O}'(k) = \mathbf{\Phi}\mathbf{\Omega}(k)$ and upon convergence of the algorithm, the outlier vector norms are computable in terms of $\mathbf{K}$, i.e., $[\|\mathbf{o}_1(\infty)\|_2^2, \ldots, \|\mathbf{o}_N(\infty)\|_2^2]' = \text{diag}[\mathbf{\Omega}'(\infty)\mathbf{K}\mathbf{\Omega}(\infty)]$. These are critical to determine the robustification paths needed to carry out the outlier sparsity control methods in Section 1.3.1. Moreover, the principal component corresponding to any given new data point $\mathbf{x}$ is obtained through the projection $\mathbf{s} = \mathbf{U}(\infty)'[\boldsymbol{\phi}(\mathbf{x}) - \mathbf{m}(\infty)] = \mathbf{\Upsilon}'(\infty)\mathbf{\Phi}'\boldsymbol{\phi}(\mathbf{x}) - \mathbf{\Upsilon}'(\infty)\mathbf{K}\boldsymbol{\mu}(\infty)$, which is again computable after N evaluations the kernel function $K_{\mathcal{H}}$.

1.5.1 Unveiling communities in social networks

Next, robust KPCA is used to identify communities and outliers in a social network of $N = 115$ college football teams, by capitalizing on the connection between KPCA and spectral clustering [HTF09, p. 548]. Nodes in the network graph represent teams belonging to eleven conferences (plus five independent teams), whereas (unweighted) edges joining pairs of nodes indicate that both teams played against each other during the Fall 2000 Division I season [GN02]. The kernel matrix used to run robust KPCA is $\mathbf{K} = \zeta\mathbf{I}_N + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{A}$ and $\mathbf{D}$ denote the graph adjacency and degree matrices, respectively; while $\zeta > 0$ is chosen to render $\mathbf{K}$ positive semi-definite. The tuning parameters are chosen as $\lambda_2 = 1.297$ so that $\|\hat{\mathbf{O}}\|_0 = 10$, while $\lambda_* = 1$, and $\bar{q} = 3$. Fig. 1.4 shows the entries of $\mathbf{K}$, where rows and columns are permuted to reveal the clustering structure found by robust KPCA (after removing the outliers); see also [MG12, Fig. 6 (top)] for a depiction of the partitioned network. The quality of the clustering is assessed through the adjusted rand index (ARI) after excluding outliers [FKG11], which yielded the value 0.8967. Four of the teams deemed as

outliers are Connecticut, Central Florida, Navy, and Notre Dame, which are indeed teams not belonging to any major conference. The community structure of traditional powerhouse conferences such as Big Ten, Big 12, ACC, Big East, and SEC was identified exactly.

1.6 Closing Summary

Outlier-robust PCA methods were developed in this chapter, to obtain low-dimensional representations of (corrupted) data. Bringing together the seemingly unrelated fields of robust statistics and sparse recovery, the surveyed robust PCA framework was found rooted at the crossroads of outlier-resilient estimation, learning via (group-) Lasso and kernel methods, and decentralized as well as real-time adaptive signal processing. Social network analysis, video surveillance, and psychometrics, were highlighted as relevant application domains.

References

1. L. Balzano. On GROUSE and incremental SVD. In *Proc. of 5th Workshop on Comp. Advances in Multi-Sensor Adaptive Proc.*, St. Martin, December 2013.
2. D. P. Bertsekas. *Nonlinear Programming*. Athena-Scientific, second edition, 1999.
3. S. Burer and R. D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, December 2005.
4. L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Proc. of Allerton Conference on Communication, Control, and Computing*, Monticello, USA, September 2010.
5. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learning*, 3:1–122, 2010.
6. L. Balzano and S. J. Wright. Local convergence of an algorithm for subspace identification from partial data. *arXiv preprint arXiv:1306.3391*, 2013.
7. N. A. Campbell. Robust procedures in multivariate analysis i: Robust covariance estimation. *Applied Stat.*, 29:231–237, 1980.
8. Y. Chi, Y. C. Eldar, and R. Calderbank. PETRELS: Parallel subspace estimation and tracking using recursive least squares from partial observations. *IEEE Trans. Signal Process.*, 61(23):5947–5959, November 2013.
9. E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58, March 2011.
10. V. Chandrasekaran, S. Sanghavi, P. A. Parillo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21:572–596, 2011.
11. E. J. Candes, M. B. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimzation. *Journal of Fourier Analysis and Applications*, 14:877–905, December 2008.
12. X. Ding, L. He, and L. Carin. Bayesian robust principal component analysis. *IEEE Trans. Image Process.*, 20, 2011.
13. F. de la Torre and M. J. Black. A framework for robust subspace learning. *Int. Jrnl. of Computer Vision*, 54:1183–209, 2003.
14. W. Dai, O. Milenkovic, and E. Kerman. Subspace evolution and transfer (SET) for low-rank matrix completion. *IEEE Trans. Signal Process.*, 59(7):3120–3132, July 2011.
15. M. Fazel, H. Hindi, and S. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proc. of the American Control Conf.*, pages 2156–2162, Denver, CO, June 2003.
16. P. Forero, V. Kekatos, and G. B. Giannakis. Outlier-aware robust clustering. In *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, pages 2244–2247, Prague, Czech Republic, May 2011.
17. J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Stat. Assoc.*, 96:1348–1360, 2001.
18. J. J. Fuchs. An inverse problem approach to robust regression. In *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, pages 180–188, Phoeniz, AZ, March 1999.
19. G. B. Giannakis, G. Mateos, S. Farahmand, V. Kekatos, and H. Zhu. USPACOR: Universal sparsity-controlling outlier rejection. In *Proc. of Intl. Conf. on Acoust., Speech and Signal Proc.*, pages 1952–1955, Prague, Czech Republic, May 2011.
20. M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99:7821–7826, 2002.

21. L. R. Goldberg. The Eugene-Springfield community sample: Information available from the research participants. Technical Report vol. 48, no. 1, Oregon Research Institue, 2008.
22. J. He, L. Balzano, and A. Szlam. Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, June 2012.
23. P. J. Huber and E. Ronchetti. *Robust Statistics*. Wiley, New York, 2009.
24. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2009.
25. O. P. John, L. P. Naumann, and C. J. Soto. Paradigm shift to the integrative big-five trait taxonomy: History, measurement, and conceptual issues. In O. P. John, R. W. Robins, and L. A. Pervin, editors, *Handbook of personality: Theory and research*. Guilford Press, New York, NY, 2008.
26. I. T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2002.
27. V. Kekatos and G. B. Giannakis. From sparse signals to sparse residuals for robust sensing. *IEEE Trans. on Signal Processing*, 59:3355–3368, July 2011.
28. E. D. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer, 2009.
29. A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *Proc. of ACM SIGMETRICS*, New York, NY, July 2004.
30. J. Mairal, J. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Jrnl. of Machine Learning Research*, 11:19–60, January 2010.
31. G. Mateos and G. B. Giannakis. Robust PCA as bilinear decomposition with outlier-sparsity regularization. *IEEE Trans. Signal Process.*, 60:5176–5190, 2012.
32. M. Mardani, G. Mateos, and G. B. Giannakis. Decentralized sparsity regularized rank minimization: Applications and algorithms. *IEEE Trans. Signal Process.*, 61:5374–5388, November 2013.
33. M. Mardani, G. Mateos, and G. B. Giannakis. Dynamic anomalography: tracking network anomalies via sparsity and low rank. *IEEE J. Sel. Topics in Signal Process.*, 7(11):50–66, February 2013.
34. M. Mardani, G. Mateos, and G. B. Giannakis. Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Trans. Signal Process.*, 63:2663–2667, March 2015.
35. B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24:227–234, 1995.
36. A. T. Puig, A. Wiesel, and A. O. Hero. Multidimensional shrinkage-thresholding operator and group LASSO penalties. *IEEE Signal Process. Letters*, 18:363–366, June 2011.
37. C. Qiu and N. Vaswani. Recursive sparse recovery in large but correlated noise. In *Proc. of Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, 2011.
38. C. Qiu, N. Vaswani, B. Lois, and L. Hogben. Recursive robust PCA or recursive sparse recovery in large but structured noise. *IEEE Trans. on Info. Theory*, 60:5007–5039, August 2014.
39. B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52:471–501, 2010.
40. P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. Wiley, New York, 1987.
41. I. Ramirez, F. Lecumberry, and G. Sapiro. Universal priors for sparse modeling. In *Proc.*

of 3rd Intl. Workshop on Comp. Advances in Multi-Sensor Adapt. Process., pages 197–200, Aruba, Dutch Antilles, December 2009.

42. B. Recht and C. Re. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5:201–226, 2013.
43. K. Slavakis, G. B. Giannakis, and G. Mateos. Modeling and optimization for big data analytics. *IEEE Signal Process. Mag.*, 31:18–31, September 2014.
44. B. Scholkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. *Computation Learning Theory: Lec. Notes in Computer Science*, 2111:416–426, 2001.
45. K. Slavakis, S.-J. Kim, G. Mateos, and G. B. Giannakis. Stochastic approximation vis-a-vis online learning for big data. *IEEE Signal Process. Mag.*, 31:124–129, November 2014.
46. N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1329–1336, Vancouver, Canada, December 2004.
47. M. Storer, P. M. Roth, M. Urschler, and H. Bischof. Fast-robust PCA. *Image Analysis: Lec. Notes in Computer Science*, 5575:430–439, 2009.
48. N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *Proc. of Learning Theory*, pages 545–560. Springer, 2005.
49. B. Scholkopf, A. J. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
50. I. Tošić and P. Frossard. Dictionary learning. *IEEE Signal Process. Mag.*, 28:27–38, March 2010.
51. J. Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Trans. on Info. Theory*, 51:1030–1051, March 2006.
52. P. Tseng. Convergence of block coordinate descent method for nondifferentiable maximization. *J. Optim. Theory Appl.*, 109:473–492, 2001.
53. H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Trans. on Info. Theory*, 58:3047–3064, May 2012.
54. L. Xu and A. L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. Neural Nets.*, 6:131–143, January 1995.
55. B. Yang. Projection approximation subspace tracking. *IEEE Trans. Signal. Process.*, 43:95–107, January 1995.
56. J. F. Yang and M. Kaveh. Adaptive eigensubspace algorithms for direction or frequency estimation and tracking. *IEEE Trans. Acoust., Speech, Signal Process.*, 36(2):241–251, February 1988.
57. M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Royal. Statist. Soc B*, 68:49–67, 2006.
58. H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Jrnl. of Comp. and Graphical Statistics*, 15(2):265–286, 2006.
59. Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma. Stable principal component pursuit. In *Proc. of Intl. Symp. on Information Theory*, pages 1518–1522, Austin, TX, June 2010.