Robust Clustering Using Outlier-Sparsity Regularization

Pedro A. Forero, *Student Member, IEEE*, Vassilis Kekatos, *Member, IEEE*, and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—Notwithstanding the popularity of conventional clustering algorithms such as K-means and probabilistic clustering, their clustering results are sensitive to the presence of outliers in the data. Even a few outliers can compromise the ability of these algorithms to identify meaningful hidden structures rendering their outcome unreliable. This paper develops robust clustering algorithms that not only aim to cluster the data, but also to identify the outliers. The novel approaches rely on the infrequent presence of outliers in the data, which translates to sparsity in a judiciously chosen domain. Leveraging sparsity in the outlier domain, outlier-aware robust K-means and probabilistic clustering approaches are proposed. Their novelty lies on identifying outliers while effecting sparsity in the outlier domain through carefully chosen regularization. A block coordinate descent approach is developed to obtain iterative algorithms with convergence guarantees and small excess computational complexity with respect to their non-robust counterparts. Kernelized versions of the robust clustering algorithms are also developed to efficiently handle high-dimensional data, identify nonlinearly separable clusters, or even cluster objects that are not represented by vectors. Numerical tests on both synthetic and real datasets validate the performance and applicability of the novel algorithms.**

***Index Terms*—(Block) coordinate descent, clustering, expectation-maximization algorithm, group-Lasso, K-means, kernel methods, mixture models, robustness, sparsity.**

I. Introduction

CLUSTERING aims to partition a set of data into subsets, called clusters, such that data assigned to the same cluster are similar in some sense. Working with unlabeled data and under minimal assumptions makes clustering a challenging, yet universal tool for revealing data structures in a gamut of applications such as DNA microarray analysis and bioinformatics, (social) network analysis, image processing, and data mining [18], [35]. Moreover, clustering can serve as a pre-processing step for supervised learning in applications where labeling data one-at-a-time is costly. Multiple interpretations across disciplines of what a cluster is, have led to an abundance of application-specific algorithms [35].

Among the algorithms which cluster data represented by vectors, K-means and Gaussian mixture model (GMM-)based clustering are two popular schemes [26], [35]. K-means relies on the Euclidean distance as a similarity measure, thereby yielding partitions that minimize the within-cluster scatter [18]. Contrastingly, soft (a.k.a. fuzzy) K-means is well-suited for overlapping clusters by allowing each datum to belong to multiple clusters [2]. GMM-based clustering considers data drawn from a probability density function (pdf), where each class-conditional pdf corresponds to a cluster [35]. Clustering then arises as a by-product of a maximum likelihood (ML) estimation framework for the GMM parameters, which are typically obtained through the expectation-maximization (EM) algorithm [11]. Kernel methods have been devised to enable clustering of nonlinearly separable clusters [29], [30].

Notwithstanding their popularity, K-means and GMM-based clustering are sensitive to inconsistent data, termed outliers, due to their functional dependency on the Euclidean distance [20]. Outliers appear infrequently in the data, emerging either due to reading errors or because they belong to rarely-seen and hence, markedly informative phenomena. However, even a few outliers can render clustering unreliable: cluster centers and model parameter estimates can be severely biased, and thus the data-to-cluster assignment is deteriorated. This motivates robustifying clustering approaches against outliers at affordable computational complexity in order to unravel the underlying structure in the data.

Several robust clustering approaches have been investigated [16]. Those more relevant to the framework developed here include possibilistic clustering, which builds on fuzzy K-means by measuring the so-called typicality of each datum with respect to each cluster to decide whether a datum is an outlier [24], [28]. However, possibilistic clustering is sensitive to initialization, and can output the same cluster more than once. Similar to [19], the noise clustering method of [10] introduces an additional cluster intended to capture all outliers, and its centroid is heuristically assumed to be equidistant from all non-outlying data. To mitigate centroid bias, the α-cut method performs K-means steps, but cluster centroids are estimated using only the α-percentage of the data assigned to each cluster [36].

Other robust alternatives include sequential clustering approaches which identify a single cluster at a time, and remove its points from the dataset [22], [38]. A minimum-volume ellipsoid containing a predetermined fraction of the data is identified per step in [22]; while [38] combines Huber's ϵ-contaminated model with a GMM [20]. However, sequentially removing points can hinder the underlying data structure.

Manuscript received July 18, 2011; revised January 04, 2012; accepted April 05, 2012. Date of publication April 26, 2012; date of current version July 10, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Raviv Raich. The work in this paper was supported in part by the AFOSR MURI Grant FA9550-10-1-0567. The work of V. Kekatos was supported by a Marie Curie International Outgoing Fellowship within the 7th European Community Framework Programme (No. 234914).

P. A. Forero and G. B. Giannakis are with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: forer002@umn.edu; georgios@umn.edu).

V. Kekatos is with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN 55455 USA, and also with the Computer Engineering and Informatics Department, University of Patras, Greece (e-mail:kekatos@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2012.2196696

Inspired by robust statistics, clustering methods based on the ℓ_1-distance (K-medians), Tukey's biweighted function, and trimmed means have been also proposed [4], [15], [23]; but they are all limited to linearly separable clusters. A clustering approach identifying clusters of arbitrary shape using kernel functions was developed in [1]. Even though resilient to outliers, this method targets density estimation, while the number of clusters identified depends critically on a grid search over a kernel parameter. Robust GMM-based clustering approaches introduce outlier-aware pdfs, and the ML problem arising is typically solved via EM-like algorithms [27], [31].

The first contribution of the present work is to introduce a data model for clustering that explicitly accounts for outliers via a deterministic outlier vector per datum (Section II). A datum is deemed an outlier if its corresponding outlier vector is nonzero. Translating the fact that outliers are rare to *sparsity* in the outlier vector domain leads to a neat connection between clustering and the *compressed sensing* (CS) paradigm [7]. Building on this model, an outlier-aware clustering methodology is developed for clustering both from the deterministic (K-means), and the probabilistic (GMMs) perspectives.

The second contribution of this work comprises various iterative clustering algorithms developed for robust hard K-means, soft K-means, and GMM-based clustering (Section III). The algorithms are based on a block coordinate descent (BCD) iteration and yield closed-form updates for each set of optimization variables. In particular, estimating the outliers boils down to solving a group-Lasso problem [37], whose solution is computed in closed form. The novel robust clustering algorithms operate at an affordable computational complexity of the same order as that of their non-robust counterparts.

Several contemporary applications in bioinformatics, (social) network analysis, image processing, and machine learning call for outlier-aware clustering of high-dimensional data, or involve nonlinearly separable clusters. To accommodate these clustering needs, the novel robust clustering algorithms are kernelized in Section IV; and this is the third contribution of our work. The assumed model not only enables such a kernelization for both K-means and the probabilistic setups, but it also yields iterative algorithms with closed-form updates. In Section V, the algorithms developed are tested using synthetic as well as real datasets from handwritten digit recognition systems and social networks. The results corroborate the effectiveness of the methods. Conclusions are drawn in Section VI.

Notation: Lower-(upper-)case boldface letters stand for column vectors (matrices), and calligraphic letters for sets; $(\cdot)^T$ denotes transposition; $\mathbb{N}_N$ the set of naturals $\{1,\ldots,N\}$; $\mathbf{0}_p$ ($\mathbf{1}_p$) the $p\times 1$ vector of all zeros (ones); $\mathbf{I}_p$ the $p\times p$ identity matrix; $\mathrm{diag}(x_1,\ldots,x_p)$ a $p\times p$ diagonal matrix with diagonal entries $x_1,\ldots,x_p$; $\mathrm{range}(\mathbf{X})$ the range space of matrix $\mathbf{X}$; $\mathbb{E}[\cdot]$ the expectation operator; $\mathcal{N}(\mathbf{x};\mathbf{m},\mathbf{\Sigma})$ the multivariate Gaussian pdf with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$ evaluated at $\mathbf{x}$; $\|\mathbf{x}\|_{\mathbf{A}} := \sqrt{\mathbf{x}^T\mathbf{A}\mathbf{x}}$ for a positive semidefinite matrix $\mathbf{A}$; $\|\mathbf{x}\|_p := (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$, with $p\geq 1$, for the ℓ_p-norm in $\mathbb{R}^n$.

II. Sparsity-Aware Clustering: Context and Criteria

After reviewing the clustering task, a model pertinent to outlier-contaminated data is introduced next. Building on this model, robust approaches are developed for K-means (Section II-A), and probabilistic clustering (Section II-B).

A. K-Means Clustering

Given a set of p-dimensional vectors $\mathcal{X} := \{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$, let $\{\mathcal{X}_1,\ldots,\mathcal{X}_C\}$ be a *partition* of $\mathcal{X}$ to C subsets (clusters) $\mathcal{X}_c \subset \mathcal{X}$ for $c\in\mathbb{N}_C$, which are collectively exhaustive, mutually exclusive, and non-empty. Partitional clustering seeks a partition of $\mathcal{X}$ such that two vectors assigned to the same cluster are closer to each other in some well-defined sense, such as the Euclidean distance, than to vectors assigned to other clusters.

Among partitional clustering methods, K-means is one of the most widely used with well-documented merits and a long history [5]. In the K-means setup, a centroid $\mathbf{m}_c\in\mathbb{R}^p$ is introduced per cluster $\mathcal{X}_c$. Then, instead of comparing distances between pairs of points in $\mathcal{X}$, the point-centroid distances $\|\mathbf{x}_n-\mathbf{m}_c\|_2$ are considered. Moreover, for each input vector $\mathbf{x}_n$, K-means introduces the unknown memberships u_{nc} for $c\in\mathbb{N}_C$, defined to be 1 when $\mathbf{x}_n\in\mathcal{X}_c$, and 0 otherwise. To guarantee a valid partition, the membership coefficients apart from being binary (c1): $u_{nc}\in\{0,1\}$; they should also satisfy the constraints (c2): $\sum_{n=1}^N u_{nc}>0$, for all c, to preclude empty clusters; and (c3): $\sum_{c=1}^C u_{nc}=1$, for all n, so that each vector is assigned to a cluster.

The K-means clustering task can be then posed as that of finding the centroids $\{\mathbf{m}_c\}_{c=1}^C$ and the cluster assignments u_{nc}'s by solving the optimization problem

$$\min_{\{\mathbf{m}_c\},\{u_{nc}\}} \sum_{n=1}^N\sum_{c=1}^C u_{nc}\|\mathbf{x}_n-\mathbf{m}_c\|_2^2 \text{ subject to (c1)–(c3).} \quad (1)$$

However, problem (1) is known to be NP-hard, even for $C=2$ [9]. Practically, a suboptimal solution is pursued using the celebrated K-means algorithm. This algorithm drops the (c2) constraint, which is checked in a post-processing step instead. Then, it alternately minimizes the cost in (1) with respect to one set of variables $\{\mathbf{m}_c\}$ or $\{u_{nc}\}$, while keeping the other one fixed, and iterates. K-means iterations are guaranteed to converge to a stationary point of (1)[32].

To gain more insight on K-means clustering, it is instructive to postulate a pertinent data model $\mathbf{x}_n=\sum_{c=1}^C u_{nc}\mathbf{m}_c+\mathbf{v}_n$, where $\mathbf{v}_n$ is a zero-mean vector capturing the deviation of $\mathbf{x}_n$ from its associated centroid $\mathbf{m}_c$. It is easy to see that under (c1)-(c3), the minimizers of (1) offer merely a blind least-squares (LS) fit of the data $\{\mathbf{x}_n\}_{n=1}^N$ respecting the cluster assignment constraints. However, such a simplistic, yet widely applicable model, does not take into account *outliers*; that is points $\mathbf{x}_n$ violating the assumed model. This fact paired with the sensitivity of the LS cost to large residuals explain K-means' vulnerability to outliers [10].

To robustify K-means, consider the following data model which explicitly accounts for outliers:

$$\mathbf{x}_n=\sum_{c=1}^C u_{nc}\mathbf{m}_c+\mathbf{o}_n+\mathbf{v}_n,\ n\in\mathbb{N}_N \quad (2)$$

where the outlier vector $\mathbf{o}_n$ is defined to be deterministically nonzero if $\mathbf{x}_n$ corresponds to an outlier, and $\mathbf{0}_p$ otherwise. The unknowns $\{u_{nc},\mathbf{m}_c,\mathbf{o}_n\}$ in (2) can now be estimated using the LS approach as the minimizers of $\sum_{n=1}^N\|\mathbf{x}_n-\sum_{c=1}^C u_{nc}\mathbf{m}_c-\mathbf{o}_n\|_2^2$, or due to (c1) and (c3), as the minimizers of $\sum_{n=1}^N\sum_{c=1}^C u_{nc}\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\|_2^2$, which are the maximum likelihood (ML) estimates if $\mathbf{v}_n\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I}_p)$.

Even if u_{nc}'s were known, estimating $\{\mathbf{m}_c\}$ and $\{\mathbf{o}_n\}$ based solely on $\{\mathbf{x}_n\}$ would be an under-determined problem. The key observation here is that most of the $\mathbf{o}_n$'s are zero. This motivates the following criterion for clustering and identification of at most $s \leq N$ outliers

$$\begin{aligned} \min_{\mathbf{M},\mathbf{O},\mathbf{U}\in\mathcal{U}_1} \quad & \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}\left\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\right\|_2^2 \\ \text{s. to} \quad & \sum_{n=1}^{N} \mathrm{I}\left(\|\mathbf{o}_n\|_2>0\right)\leq s \end{aligned} \tag{3}$$

where $\mathbf{M}:=[\mathbf{m}_1\ \cdots\ \mathbf{m}_C]$, $\mathbf{O}:=[\mathbf{o}_1\ \cdots\ \mathbf{o}_N]$, $\mathbf{U}\in\mathbb{R}^{N\times C}$ denotes the membership matrix with entries $[\mathbf{U}]_{n,c}:=u_{nc}$, $\mathcal{U}_1$ is the set of all $\mathbf{U}$ matrices satisfying (c1) and (c3), and $\mathrm{I}(\cdot)$ denotes the indicator function. Since problem (3) reduces to the K-means problem in (1) for $s=0$, the former inherits the NP-hardness of the latter. Consider now the Lagrangian form of (3)

$$\min_{\substack{\mathbf{M},\mathbf{O},\\ \mathbf{U}\in\mathcal{U}_1}} \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}\left\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\right\|_2^2+\lambda\sum_{n=1}^{N}\mathrm{I}\left(\|\mathbf{o}_n\|_2>0\right) \tag{4}$$

where $\lambda\geq 0$ is an outlier-controlling parameter. For $\lambda=0$, setting $\mathbf{o}_n=\mathbf{x}_n-\mathbf{m}_c$ for some c yields a zero optimum cost, where all $\mathbf{x}_n$'s are declared as outliers. For λ sufficiently large, the optimum $\mathbf{o}_n$'s are zero, $\mathcal{X}$ is deemed outlier-free, and problem (4) reduces to the K-means one in (1).

Along the lines of K-means, similar iterations could be pursued for suboptimally solving (4). However, such iterations cannot provide any convergence guarantees due to the discontinuity of the indicator function at zero. Aiming at a practically feasible solver of (4), consider first that $\mathbf{U}\in\mathcal{U}_1$ is given. The optimization with respect to $\{\mathbf{M},\mathbf{O}\}$ remains non-convex due to $\sum_{n=1}^{N}\mathrm{I}(\|\mathbf{o}_n\|_2>0)$. Following the successful CS paradigm, where the ℓ_0-(pseudo)norm of a vector $\mathbf{x}\in\mathbb{R}^N$, defined as $\|\mathbf{x}\|_0:=\sum_{n=1}^{N}\mathrm{I}(|x_n|>0)$, was surrogated by its convex ℓ_1-norm $\|\mathbf{x}\|_1$, the problem in (4) is replaced by

$$\min_{\mathbf{M},\mathbf{O},\mathbf{U}\in\mathcal{U}_1} \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}\left\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\right\|_2^2+\lambda\sum_{n=1}^{N}\|\mathbf{o}_n\|_2\,. \tag{5}$$

The proposed robust K-means approach is to minimize (5), which is convex in $\{\mathbf{M},\mathbf{O}\}$, but remains jointly non-convex. The algorithm for suboptimally solving the non-convex problem in (5) is postponed for Section III-A. Note that the minimization in (5) resembles the group Lasso criterion used for recovering a block-sparse vector in a linear regression setup [37]. This establishes an interesting link between robust clustering and CS. Two remarks are now in order.

Remark 1 (Colored Noise): If the covariance matrix of $\mathbf{v}_n$ in (2) is known, say $\mathbf{\Sigma}$, the ℓ_2-norms in (5) can be replaced by the weighted norms $\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}}^2$ and $\|\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}}$, respectively.

Remark 2 (ℓ_1-Penalty for Entry-Wise Outliers): The regularization term $\sum_{n=1}^{N}\|\mathbf{o}_n\|_2$ in (5) enables identifying whole data vectors as outliers. Replacing it by $\sum_{n=1}^{N}\|\mathbf{o}_n\|_1$ enables recovery of outlying data entries instead of the whole vector. Iterative solvers for this case can be developed using the methodology presented in Section III; due to space limitations this case is not pursued here.

Constraints (c1) and (c3) in (1) entail *hard* membership assignments, meaning that each vector is assigned to a single cluster. However, *soft* clustering which allows each vector to partially belong to several clusters, can better identify overlapping clusters [2]. One way to obtain fractional memberships is via soft K-means. Soft K-means differs from hard K-means by i) relaxing the binary-alphabet constraint (c1) to the box constraint (c4): $u_{nc}\in[0,1]$; and ii) by raising the u_{nc}'s in (1) to the qth power, where $q>1$ is a tuning parameter [2]. The robust soft K-means scheme proposed here amounts to replace $\mathbf{x}_n$ with its outlier-compensated version $(\mathbf{x}_n-\mathbf{o}_n)$, and leverage the sparsity of the $\mathbf{o}_n$'s. These steps lead to the following criterion:

$$\min_{\mathbf{M},\mathbf{O},\mathbf{U}\in\mathcal{U}_2} \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}^{q}\left(\left\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\right\|_2^2+\lambda\left\|\mathbf{o}_n\right\|_2\right) \tag{6}$$

where $\mathcal{U}_2$ is the set of all $\mathbf{U}$ matrices satisfying (c3)-(c4). An algorithm for approximately solving (6) is presented in Section III-A. Note that a hard partition of $\mathcal{X}$ can still be obtained from the soft u_{nc} by assigning $\mathbf{x}_n$ to the $\hat{c}$th cluster, where $\hat{c}:=\arg\max_c u_{nc}$.

B. Probabilistic Clustering

An alternative way to perform soft clustering is by following a probabilistic approach [35]. To this end, a mixture distribution model is postulated for $\mathbf{x}_n$, while $\{u_{nc}\}_{c=1}^{C}$ are now interpreted as unobserved (latent) random variables. The centroids $\{\mathbf{m}_c\}_{c=1}^{C}$ are treated as deterministic parameters of the mixture distribution, and their ML estimates are subsequently obtained via the EM algorithm.

To account for outliers, probabilistic clustering is generalized to model (2). Suppose that the $\{\mathbf{x}_n\}$'s in (2) are i.i.d. drawn from a mixture model where the $\{\mathbf{o}_n\}$'s are deterministic parameters. The memberships $\mathbf{u}_n:=[u_{n1}\cdots u_{nC}]^T$ are latent random vectors, corresponding to the rows of $\mathbf{U}$, and take values in $\{\mathbf{e}_1,\ldots,\mathbf{e}_C\}$, where $\mathbf{e}_c$ is the cth column of $\mathbf{I}_C$. If $\mathbf{x}_n$ is drawn from the cth mixture component, then $\mathbf{u}_n=\mathbf{e}_c$. Assume further that the class-conditional pdf's are Gaussian and modeled as $p(\mathbf{x}_n|\mathbf{u}_n=\mathbf{e}_c)=\mathcal{N}(\mathbf{x}_n;\mathbf{m}_c+\mathbf{o}_n,\mathbf{\Sigma})$ for all n and c. This implies that $p(\mathbf{x}_n)=\sum_{c=1}^{C}\pi_c\,\mathcal{N}(\mathbf{x}_n;\mathbf{m}_c+\mathbf{o}_n,\mathbf{\Sigma})$ with $\pi_c:=\Pr(\mathbf{u}_n=\mathbf{e}_c)$. If the $\mathbf{x}_n$'s are independent, the log-likelihood of the input data is

$$L(\mathbf{X};\boldsymbol{\pi},\mathbf{M},\mathbf{O},\mathbf{\Sigma}):=\sum_{n=1}^{N}\log\left(\sum_{c=1}^{C}\pi_c\mathcal{N}\left(\mathbf{x}_n;\mathbf{m}_c+\mathbf{o}_n,\mathbf{\Sigma}\right)\right) \tag{7}$$

where $\mathbf{X}:=[\mathbf{x}_1\ \cdots\ \mathbf{x}_N]$, and $\boldsymbol{\pi}:=[\pi_1\cdots\pi_C]^T$. Controlling the number of outliers (number of zero $\mathbf{o}_n$ vectors) suggests minimizing the *regularized* negative log-likelihood as

$$\min_{\mathbf{\Theta}} -L\left(\mathbf{X};\mathbf{\Theta}\right)+\lambda\sum_{n=1}^{N}\|\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}} \tag{8}$$

where $\mathbf{\Theta}:=\{\boldsymbol{\pi}\in\mathcal{P},\mathbf{M},\mathbf{O},\mathbf{\Sigma}\succ\mathbf{0}\}$ is the set of all model parameters, $\mathcal{P}$ is the probability simplex $\mathcal{P}:=\{\boldsymbol{\pi}:\boldsymbol{\pi}^T\mathbf{1}=1\ \text{and}\ \boldsymbol{\pi}\geq\mathbf{0}\}$, and $\mathbf{\Sigma}\succ\mathbf{0}$ means that $\mathbf{\Sigma}$ is a positive definite

matrix. An EM-based solver of (8) is derived in Section III-B. Having estimated the likelihood parameters, the posterior probabilities $\gamma_{nc} := \Pr(\mathbf{u}_n = \mathbf{e}_c|\mathbf{x}_n)$ can be readily obtained and interpreted as soft memberships.

Although having a common covariance $\mathbf{\Sigma}$ $\forall c$ may seem restrictive, it guarantees that the GMM is well-posed, thereby avoiding spurious unbounded likelihood values [3, p. 433]. Specifically, it is easy to see that even if all $\mathbf{o}_n$'s are set to zero, the log-likelihood of a GMM with different $\mathbf{\Sigma}_c$ per mixture grows unbounded, e.g., by setting one of the $\mathbf{m}_c$'s equal to an $\mathbf{x}_n$ and letting $\mathbf{\Sigma}_c \to \mathbf{0}$ for that particular c. This possibility for unboundedness is also present in (8), and justifies the use of a common $\mathbf{\Sigma}$. But even with a common covariance, vectors $\mathbf{o}_n$ can drive the log-likelihood in (7) to infinity: consider for example, any $(\mathbf{m}_c, \mathbf{o}_n)$ pair satisfying $\mathbf{x}_n = \mathbf{m}_c + \mathbf{o}_n$, and let $\mathbf{\Sigma} \to \mathbf{0}$. To make the problem of maximizing $L(\mathbf{X}; \mathbf{\Theta})$ well-posed and in analogy to the deterministic setup (cf. Remark 1), the $\|\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}}$ regularizer is introduced. Note also that for $\lambda \to \infty$, the optimal $\mathbf{O}$ is zero and (8) reduces to the conventional MLE estimation of a GMM; whereas for $\lambda \to 0$, the cost in (8) becomes unbounded from below.

III. Robust Clustering Algorithms

Algorithms for solving the problems formulated in Section II are developed here. Section III-A focuses on the minimization of (6), while the minimization in (5) is obtained from (6) for $q = 1$. In Section III-B, an algorithm for minimizing (8) is derived based on the EM approach. Finally, modified versions of the new algorithms with enhanced resilience to outliers are pursued in Section III-C.

A. Robust (Soft) K-Means Algorithms

Consider first solving (6) for $q > 1$. Although the cost is jointly nonconvex, it is convex with respect to each of $\mathbf{M}$, $\mathbf{O}$, and $\mathbf{U}$. To develop a suboptimum yet practical solver, this per-variable convexity motivates a BCD algorithm, which minimizes the cost iteratively with respect to each optimization variable while holding the other two variables fixed. Let $\mathbf{M}^{(t)}$, $\mathbf{O}^{(t)}$, and $\mathbf{U}^{(t)}$ denote the solutions found at the tth iteration. Also, initialize $\mathbf{U}^{(0)}$ randomly in $\mathcal{U}_2$, and $\mathbf{O}^{(0)}$ to zero.

In the first step of the tth iteration, (6) is optimized over $\mathbf{M}$ for $\mathbf{U} = \mathbf{U}^{(t-1)}$ and $\mathbf{O} = \mathbf{O}^{(t-1)}$. The optimization decouples over the $\mathbf{m}_c$'s, and every $\mathbf{m}_c^{(t)}$ is the closed-form solution of an LS problem as

$$\mathbf{m}_c^{(t)} = \frac{\sum_{n=1}^{N} (u_{nc}^{(t-1)})^q \left(\mathbf{x}_n - \mathbf{o}_n^{(t-1)}\right)}{\sum_{n=1}^{N} (u_{nc}^{(t-1)})^q}. \tag{9}$$

In the second step, the task is to minimize (6) with respect to $\mathbf{O}$ for $\mathbf{U} = \mathbf{U}^{(t-1)}$ and $\mathbf{M} = \mathbf{M}^{(t)}$. The optimization problem decouples per index n, so that each $\mathbf{o}_n$ can be found as the minimizer of

$$\phi^{(t)}(\mathbf{o}_n) := \sum_{c=1}^{C} (u_{nc}^{(t-1)})^q \left(\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n\|_2^2 + \lambda \|\mathbf{o}_n\|_2\right). \tag{10}$$

The cost $\phi^{(t)}(\mathbf{o}_n)$ is convex but non-differentiable. However, its minimizer can be expressed in closed form as shown in the ensuing proposition.

Proposition 1: *The optimization problem in (10) is uniquely minimized by*

$$\mathbf{o}_n^{(t)} = \mathbf{r}_n^{(t)} \left[1 - \frac{\lambda}{2\|\mathbf{r}_n^{(t)}\|_2}\right]_+ \tag{11}$$

where $[x]_+ := \max\{x, 0\}$, *and* $\mathbf{r}_n^{(t)}$ *is defined as*

$$\mathbf{r}_n^{(t)} := \frac{\sum_{c=1}^{C} (u_{nc}^{(t-1)})^q \left(\mathbf{x}_n - \mathbf{m}_c^{(t)}\right)}{\sum_{c=1}^{C} (u_{nc}^{(t-1)})^q}. \tag{12}$$

Proof: See Appendix A. ∎

The update for $\mathbf{o}_n^{(t)}$ in (11) reveals two interesting points: i) the cost $\phi^{(t)}(\mathbf{o}_n)$ indeed favors zero minimizers; and ii) the number of outliers is controlled by λ. After updating vector $\mathbf{r}_n^{(t)}$, its norm is compared against the threshold $\frac{\lambda}{2}$. If $\|\mathbf{r}_n^{(t)}\|_2$ exceeds $\frac{\lambda}{2}$, vector $\mathbf{x}_n$ is deemed an outlier, and it is compensated by a nonzero $\mathbf{o}_n^{(t)}$. Otherwise, $\mathbf{o}_n^{(t)}$ is set to zero and $\mathbf{x}_n$ is clustered as a regular point.

During the last step of the tth iteration, (6) is minimized over $\mathbf{U} \in \mathcal{U}_2$ for $\mathbf{M} = \mathbf{M}^{(t)}$ and $\mathbf{O} = \mathbf{O}^{(t)}$. Similar to the conventional soft K-means, the minimizer is available in closed form as [2]

$$u_{nc}^{(t)} = \left[\sum_{c'=1}^{C} \left(\frac{\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n^{(t)}\|_2^2 + \lambda\|\mathbf{o}_n^{(t)}\|_2}{\|\mathbf{x}_n - \mathbf{m}_{c'}^{(t)} - \mathbf{o}_n^{(t)}\|_2^2 + \lambda\|\mathbf{o}_n^{(t)}\|_2}\right)^{\frac{1}{q-1}}\right]^{-1}. \tag{13}$$

Regarding the robust hard K-means, a similar BCD approach for solving (5) leads to updating $\mathbf{M}^{(t)}$ and $\mathbf{O}^{(t)}$ via (9), and (11)–(12) for $q = 1$. Updating $\mathbf{U}^{(t)}$ boils down to the minimum-distance rule

$$u_{nc}^{(t)} = \begin{cases} 1, & c = \arg\min_{c'} \|\mathbf{x}_n - \mathbf{m}_{c'}^{(t)} - \mathbf{o}_n^{(t)}\|_2 \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Note that (14) is the limit case of (13) for $q \to 1^+$.

The robust K-means (RKM) algorithm is tabulated as Algorithm 1. RKM is terminated when $\frac{\|\mathbf{M}^{(t)} - \mathbf{M}^{(t-1)}\|_F}{\|\mathbf{M}^{(t)}\|_F} \leq \epsilon_s$, where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and ϵ_s is a small positive threshold, e.g., $\epsilon_s = 10^{-6}$. The computational resources needed by RKM are summarized next.

Algorithm 1: Robust K-means

Require: Input data matrix $\mathbf{X}$, number of clusters C, $q \geq 1$, and $\lambda > 0$.

1: Initialize $\mathbf{O}^{(0)}$ to zero and $\mathbf{U}^{(0)}$ randomly in $\mathcal{U}_2$.
2: **for** $t = 1, 2, \ldots$ **do**
3: Update $\mathbf{M}^{(t)}$ via (9).
4: Update $\mathbf{O}^{(t)}$ via (11)–(12).
5: Update $\mathbf{U}^{(t)}$ via (13) ($q > 1$) or (14) ($q = 1$).
6: **end for**

Remark 3 (Computational Complexity of RKM): Suppose for concreteness that: (as1) the number of clusters is small, e.g., $C < p$; and (as2) the number of points is much larger than the input dimension, i.e., $N \gg p$. When (as2) does not hold, a modification of RKM is developed in Section IV. Under (as1)-(as2), the conventional K-means algorithm performs $\mathcal{O}(NCp)$ scalar operations per iteration, and requires storing $\mathcal{O}(Np)$ scalar variables. For RKM, careful counting shows that the per iteration time-complexity is maintained at $\mathcal{O}(NCp)$: (13) requires computing the NC Euclidean distances $\|\mathbf{x}_n - \mathbf{m}_c^{(t-1)} - \mathbf{o}_n^{(t-1)}\|_2^2$ and the N norms $\|\mathbf{o}_n^{(t-1)}\|_2$ which is $\mathcal{O}(NCp)$; $\mathbf{m}_c^{(t)}$'s are updated in $\mathcal{O}(NCp)$; while (11)–(12) entail $\mathcal{O}(NCp)$ operations. Further, the memory requirements of RKM are of the same order as those for K-means. Note also that the additional $N \times p$ matrix $\mathbf{O}$ can be stored using sparse structures.

The RKM iterations are convergent under mild conditions. This follows because the sequence of cost function values is non-increasing. Since the cost is bounded below, the function value sequences are guaranteed to converge. Convergence of the RKM iterations is characterized next.

Proposition 2: *The RKM algorithm for $q \geq 1$ converges to a coordinate-wise minimum of (6). Moreover, the hard RKM algorithm $(q = 1)$ converges to a local minimum of* (5).

Proof: See Appendix B. ∎

B. Robust Probabilistic Clustering Algorithm

An EM approach is developed in this subsection to carry out the minimization in (8). If $\mathbf{U}$ were known, the model parameters $\boldsymbol{\Theta}$ could be estimated by minimizing the regularized negative log-likelihood of the *complete data* $(\mathbf{X}, \mathbf{U})$; that is,

$$\min_{\boldsymbol{\Theta}} -L(\mathbf{X}, \mathbf{U}; \boldsymbol{\Theta}) + \lambda \sum_{n=1}^{N} \|\mathbf{o}_n\|_{\boldsymbol{\Sigma}^{-1}} \tag{15}$$

where

$$L(\mathbf{X}, \mathbf{U}; \boldsymbol{\Theta}) := \sum_{n=1}^{N} \sum_{c=1}^{C} u_{nc} \left(\log \pi_c + \log \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c + \mathbf{o}_n, \boldsymbol{\Sigma})\right). \tag{16}$$

However, since $\mathbf{U}$ is not observed, the cost in (15) is suboptimally minimized by iterating the two steps of the EM method. Let $\boldsymbol{\Theta}^{(t)}$ denote the model parameter values at the tth iteration. During the E-step of the tth iteration, the expectation $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(t-1)}) := \mathbb{E}_{\mathbf{U}|\mathbf{X};\boldsymbol{\Theta}^{(t-1)}}[L(\mathbf{X}, \mathbf{U}; \boldsymbol{\Theta})]$ is evaluated. Since $L(\mathbf{X}, \mathbf{U}; \boldsymbol{\Theta})$ is a linear function of $\mathbf{U}$, and u_{nc}'s are binary random variables, it follows that

$$Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(t-1)}) = \sum_{n=1}^{N} \sum_{c=1}^{C} \gamma_{nc}^{(t)} \left(\log \pi_c + \log \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c + \mathbf{o}_n, \boldsymbol{\Sigma})\right) \tag{17}$$

where $\gamma_{nc}^{(t)} := \Pr\left(\mathbf{u}_n = \mathbf{e}_c | \mathbf{x}_n; \boldsymbol{\Theta}^{(t-1)}\right)$. Using Bayes' rule, the posterior probabilities $\gamma_{nc}^{(t)}$ are evaluated in closed form as

$$\gamma_{nc}^{(t)} = \frac{\pi_c^{(t-1)} \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c^{(t-1)} + \mathbf{o}_n^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})}{\sum_{c'=1}^{C} \pi_{c'}^{(t-1)} \mathcal{N}(\mathbf{x}_n; \mathbf{m}_{c'}^{(t-1)} + \mathbf{o}_n^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})}. \tag{18}$$

During the M-step, $\boldsymbol{\Theta}^{(t)}$ is updated as

$$\boldsymbol{\Theta}^{(t)} = \arg\min_{\boldsymbol{\Theta}} -Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(t-1)}) + \lambda \sum_{n=1}^{N} \|\mathbf{o}_n\|_{\boldsymbol{\Sigma}^{-1}}. \tag{19}$$

A BCD strategy that updates each set of parameters in $\boldsymbol{\Theta}$ one at a time with all other ones fixed, is described next. First, the cost in (19) is minimized with respect to $\boldsymbol{\pi}$. Given that $\sum_{c=1}^{C} \gamma_{nc}^{(t)} = 1$ for all n, the minimizer of $-\sum_{n=1}^{N} \sum_{c=1}^{C} \gamma_{nc}^{(t)} \log \pi_c$ over $\mathcal{P}$ is found in closed form as

$$\pi_c^{(t)} = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nc}^{(t)} \quad \text{for all} \quad c \in \mathbb{N}_C. \tag{20}$$

Subsequently, (19) is minimized with respect to $\mathbf{M}$ while $\boldsymbol{\pi}$, $\mathbf{O}$, and $\boldsymbol{\Sigma}$ are set respectively to $\boldsymbol{\pi}^{(t)}$, $\mathbf{O}^{(t-1)}$, and $\boldsymbol{\Sigma}^{(t-1)}$. The centroids are updated as the minimizers of a weighted LS cost yielding

$$\mathbf{m}_c^{(t)} = \frac{\sum_{n=1}^{N} \gamma_{nc}^{(t)} \left(\mathbf{x}_n - \mathbf{o}_n^{(t-1)}\right)}{\sum_{n=1}^{N} \gamma_{nc}^{(t)}} \quad \text{for all} \quad c \in \mathbb{N}_C. \tag{21}$$

Then, (19) is minimized with respect to $\mathbf{O}$ while keeping the rest of the model parameters fixed to their already updated values. This optimization decouples over n, and one has to solve

$$\min_{\mathbf{o}_n} \sum_{c=1}^{C} \frac{\gamma_{nc}^{(t)}}{2} \|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n\|^2_{(\boldsymbol{\Sigma}^{(t-1)})^{-1}} + \lambda \|\mathbf{o}_n\|_{(\boldsymbol{\Sigma}^{(t-1)})^{-1}} \tag{22}$$

for all $n \in \mathbb{N}_N$. For a full covariance $\boldsymbol{\Sigma}$, (22) can be solved as a second-order cone program. For the case of *spherical* clusters, i.e., $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}_p$, solving (22) simplifies considerably. Specifically, the cost can then be written as $\sum_{c=1}^{C} \gamma_{nc}^{(t)} \|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n\|_2^2 + 2\lambda\sigma^{(t-1)} \|\mathbf{o}_n\|_2$, which is similar to the cost in (10) for $q = 1$, and for an appropriately scaled λ. Building on the solution of (10), the $\mathbf{o}_n$'s are updated as

$$\mathbf{o}_n^{(t)} = \mathbf{r}_n^{(t)} \left[1 - \frac{\lambda\sigma^{(t-1)}}{\|\mathbf{r}_n^{(t)}\|_2}\right]_+ \tag{23}$$

after redefining the residual vector as $\mathbf{r}_n^{(t)} := \sum_{c=1}^{C} \gamma_{nc}^{(t)} (\mathbf{x}_n - \mathbf{m}_c^{(t)})$ in lieu of (12). Interestingly, the thresholding rule of (23) shows that $\sigma^{(t-1)}$ affects the detection of outliers. In fact, in this probabilistic setting, the threshold for outlier identification is proportional to the value of the outlier-compensated standard deviation estimate and, hence, it is adapted to the empirical distribution of the data.

The M-step is concluded by minimizing (19) with respect to $\boldsymbol{\Sigma}$ for $\boldsymbol{\pi} = \boldsymbol{\pi}^{(t)}$, $\mathbf{M} = \mathbf{M}^{(t)}$, and $\mathbf{O} = \mathbf{O}^{(t)}$, i.e.,

$$\min_{\boldsymbol{\Sigma} \succ \mathbf{0}} \sum_{n=1}^{N} \sum_{c=1}^{C} \frac{\gamma_{nc}^{(t)}}{2} \|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n^{(t)}\|^2_{\boldsymbol{\Sigma}^{-1}} + \frac{N}{2} \log\det\boldsymbol{\Sigma} + \lambda \sum_{n=1}^{N} \|\mathbf{o}_n^{(t)}\|_{\boldsymbol{\Sigma}^{-1}}. \tag{24}$$

For a generic $\mathbf{\Sigma}$, (24) must be solved numerically, e.g., via gradient descent or interior point methods. Considering *spherical* clusters for simplicity, the first order optimality condition for (24) requires solving a quadratic equation in $\sigma^{(t)}$. Ignoring the negative root of this equation, $\sigma^{(t)}$ is found as

$$\sigma^{(t)} = \frac{\lambda}{2Np}\sum_{n=1}^{N}\|\mathbf{o}_n^{(t)}\|_2 + \sqrt{\sum_{n=1}^{N}\sum_{c=1}^{C}\frac{\gamma_{nc}^{(t)}}{Np}\|\mathbf{x}_n-\mathbf{m}_c^{(t)}-\mathbf{o}_n^{(t)}\|_2^2+\left(\lambda\sum_{n=1}^{N}\frac{\|\mathbf{o}_n^{(t)}\|_2}{2Np}\right)^2}. \quad (25)$$

The robust probabilistic clustering (RPC) scheme is tabulated as Algorithm 2. For spherical clusters, its complexity remains $\mathcal{O}(NCp)$ operations per iteration, even though the constants involved are larger than those in the RKM algorithm. Similar to RKM, the RPC iterations are convergent under mild conditions. Convergence of the RPC iterations is established in the next proposition.

Algorithm 2: Robust Probabilistic Clustering

Require: Input data matrix $\mathbf{X}$, number of clusters C, and parameter $\lambda > 0$.

1: Randomly initialize $\mathbf{M}^{(0)}$, $\boldsymbol{\pi}^{(0)} \in \mathcal{P}$, and set $\mathbf{\Sigma}^{(0)} = \delta \boldsymbol{I}_p$ ($\sigma^{(0)} = \sqrt{\delta}$) for $\delta > 0$, and $\mathbf{O}^{(0)}$ to zero.

2: **for** $t = 1, 2, \ldots$ **do**

3: Update $\gamma_{nc}^{(t)}$ via (18) for all $n,\ c$.

4: Update $\boldsymbol{\pi}^{(t)}$ via (20).

5: Update $\mathbf{M}^{(t)}$ via (21).

6: Update $\mathbf{O}^{(t)}$ by solving (22) ((23)).

7: Update $\mathbf{\Sigma}^{(t)}$ ($\sigma^{(t)}$) via (24) ((25)).

8: **end for**

Proposition 3: T*he RPC iterations converge to a coordinate-wise minimum of the penalized negative log-likelihood in* (8).

Proof: See Appendix C. ■

Proposition 3 guarantees that the RPC iterations converge. However, since each non-differentiable term $\|\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}}$ involves two different optimization variables $\mathbf{\Sigma}$ and $\mathbf{o}_n$, the BCD iteration can be trapped at a coordinate-wise local minimum, which is not necessarily a local minimum of (8). Once the iterations have converged, the final γ_{nc}'s can be interpreted as soft cluster assignments, whereby hard assignments can be obtained via the maximum a posteriori detection rule, i.e., $\mathbf{x}_n \in \mathcal{X}_c$ for $c = \arg\max_{c'} \gamma_{nc'}$.

Remark 4 (Selecting λ): Tuning λ is possible if additional information, e.g., on the percentage of outliers, is available. The robust clustering algorithm is run for a decreasing sequence of λ values $\{\lambda_g\}$, using "warm starts" [14], until the expected number of outliers is identified. When solving for λ_g, warm start refers to the optimization variables initialized to the solution obtained for λ_{g-1}. Hence, running the algorithm over $\{\lambda_g\}$ can be efficiently done, because few BCD iterations per λ_g suffice for convergence.

C. Weighted Robust Clustering Algorithms

As already mentioned, the robust clustering methods presented so far approximate the discontinuous penalty $\mathrm{I}(\|\mathbf{o}_n\|_2 > 0)$ by $\|\mathbf{o}_n\|_2$, mimicking the CS paradigm in which $\mathrm{I}(|x| > 0)$ is surrogated by the convex function $|x|$. However, it has been argued that non-convex functions such as $\log(|x| + \epsilon)$ for a small $\epsilon > 0$ can offer tighter approximants of $\mathrm{I}(|x| > 0)$ [34]. This rationale prompts one to replace $\|\mathbf{o}_n\|_2$ in (5), (6), and (8), with $\log(\|\mathbf{o}_n\|_2 + \epsilon)$ to further enhance block sparsity in $\mathbf{o}_n$'s, and thereby improve resilience to outliers.

Altering the regularization modifies the BCD algorithms only when minimizing with respect to $\mathbf{O}$. This particular step remains decoupled across $\mathbf{o}_n$'s, but instead of the $\phi^{(t)}(\mathbf{o}_n)$ in (10), one minimizes

$$\phi_w^{(t)}(\mathbf{o}_n) := \sum_{c=1}^{C}(u_{nc}^{(t-1)})^q \times \left(\|\mathbf{x}_n-\mathbf{m}_c^{(t)}-\mathbf{o}_n\|_2^2+\lambda\cdot\log\left(\|\mathbf{o}_n\|_2+\epsilon\right)\right) \quad (26)$$

that is no longer convex. The optimization in (26) is performed using a single iteration of the majorization-minimization (MM) approach[1] [25]. The cost $\phi_w^{(t)}(\mathbf{o}_n)$ is majorized by a function $f^{(t)}(\mathbf{o}_n;\mathbf{o}_n^{(t-1)})$, which means that $\phi_w^{(t)}(\mathbf{o}_n) \leq f^{(t)}(\mathbf{o}_n;\mathbf{o}_n^{(t-1)})$ for every $\mathbf{o}_n$ and $\phi_w^{(t)}(\mathbf{o}_n) = f^{(t)}(\mathbf{o}_n;\mathbf{o}_n^{(t-1)})$ when $\mathbf{o}_n = \mathbf{o}_n^{(t-1)}$. Then $f^{(t)}(\mathbf{o}_n;\mathbf{o}_n^{(t-1)})$ is minimized with respect to $\mathbf{o}_n$ to obtain $\mathbf{o}_n^{(t)}$.

To find a majorizer for $\phi_w^{(t)}(\mathbf{o}_n)$, the concavity of the logarithm is exploited, i.e., the fact that $\log x \leq \log x_o + \frac{x}{x_o} - 1$ for any positive x and x_o. Applying the last inequality for the penalty and ignoring the constant terms involved, we end up minimizing

$$f^{(t)}(\mathbf{o}_n;\mathbf{o}_n^{(t-1)}) := \sum_{c=1}^{C}(u_{nc}^{(t-1)})^q\left(\|\mathbf{x}_n-\mathbf{m}_c^{(t)}-\mathbf{o}_n\|_2^2+\lambda_n^{(t)}\|\mathbf{o}_n\|_2\right) \quad (27)$$

where $\lambda_n^{(t)} := \frac{\lambda}{(\|\mathbf{o}_n^{(t-1)}\|_2+\epsilon)}$. Comparing (27) to (10) shows that the new regularization results in a weighted version of the original one. The only difference between the robust algorithms presented earlier and their henceforth termed *weighted* counterparts is the definition of λ. At iteration t, larger values for $\|\mathbf{o}_n^{(t-1)}\|_2$ lead to smaller thresholds in the thresholding rules (cf. (11), (23)), thereby making $\mathbf{o}_n$ more likely to be selected as nonzero. The weighted robust clustering algorithms initialize $\mathbf{o}_n^{(0)}$ to the associated $\mathbf{o}_n$ value the non-weighted algorithm converged to. Thus, to run the weighted RKM for a specific value of λ, the RKM needs to be run first. Then, weighted RKM is run with all the variables initialized to the values attained by RKM, but with the $\lambda_n^{(1)}$ as defined earlier.

The MM step combined with the BCD algorithms developed hitherto are convergent under mild assumptions. To see this,

[1]Note that the MM approach for minimizing $\phi_w^{(t)}(\mathbf{o}_n)$ at the tth BCD iteration involves several internal MM iterations. Due to the external BCD iterations and to speed up the algorithm, a single MM iteration is performed per BCD iteration t.

note that the sequences of objective values for RKM and RPC are both non-increasing. Since the respective cost functions are bounded below, those sequences are guaranteed to converge. Characterizing the points and speed of convergence goes beyond the scope of this paper.

IV. Clustering High-Dimensional and Nonlinearly Separable Data

The robust clustering algorithms derived so far involve generally $\mathcal{O}(NCp)$ operations per iteration. However, several applications entail clustering relatively few but *high-dimensional* data in the presence of outliers. In imaging applications, one may wish to cluster $N = 500$ images of say $p = 800 \times 600 = 480{,}000$ pixels; while in DNA microarray analysis, some tens of (potentially erroneous or rarely occurring) DNA samples are to be clustered based on their expression levels over thousands of genes [18]. In such clustering scenarios where $p \gg N$, an efficient method should avoid storing and processing p-dimensional vectors [12]. To this end, the algorithms of Section III are kernelized here [29]. It will be shown that these kernelized algorithms require $\mathcal{O}(N^3C)$ operations per iteration and $\mathcal{O}(N^2)$ space; hence, they are preferable when $p > N^2$. This kernelization does not only offer processing savings in the high-dimensional data regime (cf. Section IV-A), but it also critically enables identifying nonlinearly separable clusters (cf. Section IV-B).

A. Robust K-Means for High-Dimensional Data

Without loss of generality, the focus is on kernelizing the robust soft K-means algorithm. Consider the $N \times C$ matrix $\mathbf{U}_q$ with entries $[\mathbf{U}_q]_{nc} := u_{nc}^q$, and the Gramian $\mathbf{K} := \mathbf{X}^T\mathbf{X}$ formed by all pairwise inner products between the input vectors. Even though computing $\mathbf{K}$ costs $\mathcal{O}(N^2p)$, it is performed only once. Note that the updates (9), (11), and (13) involve inner products between the p-dimensional vectors $\{\mathbf{o}_n, \mathbf{r}_n\}_{n=1}^N$, and $\{\mathbf{m}_c\}_{c=1}^C$. If $\{\mathbf{v}_i \in \mathbb{R}^p\}_{i=1}^2$ is a pair of any of these vectors, the cost for computing $\mathbf{v}_1^T\mathbf{v}_2$ is clearly $\mathcal{O}(p)$. However, if all these vectors lie in $\mathrm{range}(\mathbf{X})$, i.e., if there exist $\{\mathbf{w}_i \in \mathbb{R}^N\}_{i=1}^2$ such that $\{\mathbf{v}_i = \mathbf{X}\mathbf{w}_i\}_{i=1}^2$, then $\mathbf{v}_1^T\mathbf{v}_2 = \mathbf{w}_1^T\mathbf{K}\mathbf{w}_2$, and the inner product can be alternatively calculated in $\mathcal{O}(N^2)$. Hinging on this observation, it is first shown that all the $p \times 1$ vectors involved indeed lie in $\mathrm{range}(\mathbf{X})$. The proof is by induction: if at the $(t-1)$st iteration every $\mathbf{o}_n^{(t-1)} \in \mathrm{range}(\mathbf{X})$ and $\mathbf{U}^{(t-1)} \in \mathcal{U}_2$, it is shown that $\mathbf{o}_n^{(t)}$, $\mathbf{m}_c^{(t)}$, $\mathbf{r}_n^{(t)}$ updated by RKM lie in $\mathrm{range}(\mathbf{X})$ as well.

Suppose that at the tth iteration, the matrix $\mathbf{U}^{(t-1)}$ defining $\mathbf{U}_q^{(t-1)}$ is in $\mathcal{U}_2$, while there exists matrix $\mathbf{A}^{(t-1)}$ such that $\mathbf{O}^{(t-1)} = \mathbf{X}\mathbf{A}^{(t-1)}$. Then, the update of the centroids in (9) can be expressed as

$$\mathbf{M}^{(t)} = (\mathbf{X} - \mathbf{O}^{(t-1)})\mathbf{U}_q^{(t-1)}\mathrm{diag}^{-1}((\mathbf{U}_q^{(t-1)})^T\mathbf{1}_N) \quad (28)$$

$$= \mathbf{X}\mathbf{B}^{(t)} \quad (29)$$

where

$$\mathbf{B}^{(t)} := (\mathbf{I}_N - \mathbf{A}^{(t-1)})\mathbf{U}_q^{(t-1)}\mathrm{diag}^{-1}((\mathbf{U}_q^{(t-1)})^T\mathbf{1}_N). \quad (30)$$

Before updating $\mathbf{O}^{(t)}$, the residual vectors $\mathbf{r}_n$'s must be updated via (12). Concatenating the residuals in $\mathbf{R}^{(t)} := [\mathbf{r}_1^{(t)} \ldots \mathbf{r}_N^{(t)}]$, the update in (12) can be rewritten in matrix form as

$$\mathbf{R}^{(t)} = \mathbf{X} - \mathbf{M}^{(t)}(\mathbf{U}_q^{(t-1)})^T\mathrm{diag}^{-1}(\mathbf{U}_q^{(t-1)}\mathbf{1}_C) \quad (31)$$

$$= \mathbf{X}\boldsymbol{\Delta}^{(t)} \quad (32)$$

where

$$\boldsymbol{\Delta}^{(t)} := \mathbf{I}_N - \mathbf{B}^{(t)}(\mathbf{U}_q^{(t-1)})^T\mathrm{diag}^{-1}(\mathbf{U}_q^{(t-1)}\mathbf{1}_C). \quad (33)$$

From (11), every $\mathbf{o}_n^{(t)}$ is a scaled version of $\mathbf{r}_n^{(t)}$ and the scaling depends on $\|\mathbf{r}_n^{(t)}\|_2$. Based on (31), the latter can be readily computed as $\|\mathbf{r}_n^{(t)}\|_2 = \sqrt{(\boldsymbol{\delta}_n^{(t)})^T\mathbf{K}\boldsymbol{\delta}_n^{(t)}} = \|\boldsymbol{\delta}_n^{(t)}\|_{\mathbf{K}}$, where $\boldsymbol{\delta}_n^{(t)}$ stands for the nth column of $\boldsymbol{\Delta}^{(t)}$. Upon applying the thresholding operator, one arrives at the update

$$\mathbf{O}^{(t)} = \mathbf{X}\mathbf{A}^{(t)} \quad (34)$$

where the nth column of $\mathbf{A}^{(t)}$ is given by

$$\boldsymbol{\alpha}_n^{(t)} = \boldsymbol{\delta}_n^{(t)}\left[1 - \frac{\lambda}{2\|\boldsymbol{\delta}_n^{(t)}\|_{\mathbf{K}}}\right]_+, \quad \forall n. \quad (35)$$

Having proved the inductive step by (34), the argument is complete if and only if the outlier variables $\mathbf{O}$ are initialized as $\mathbf{O}^{(0)} = \mathbf{X}\mathbf{A}^{(0)}$ for some $\mathbf{A}^{(0)}$, including the practically interesting and meaningful initialization at zero. The result just proved can be summarized as follows.

Proposition 4: *By choosing* $\mathbf{O}^{(0)} = \mathbf{X}\mathbf{A}^{(0)}$ *for any* $\mathbf{A}^{(0)} \in \mathbb{R}^{N\times N}$ *and* $\mathbf{U}^{(0)} \in \mathcal{U}_2$, *the columns of the matrix variables* $\mathbf{O}$, $\mathbf{M}$, *and* $\mathbf{R}$ *updated by RKM all lie in* $\mathrm{range}(\mathbf{X})$*; i.e., there exist known* $\mathbf{A}^{(t)}$, $\mathbf{B}^{(t)}$, *and* $\boldsymbol{\Delta}^{(t)}$, *such that* $\mathbf{O}^{(t)} = \mathbf{X}\mathbf{A}^{(t)}$, $\mathbf{M}^{(t)} = \mathbf{X}\mathbf{B}^{(t)}$, *and* $\mathbf{R}^{(t)} = \mathbf{X}\boldsymbol{\Delta}^{(t)}$ *for all* t.

What remains to be kernelized are the updates for the cluster assignments. For the update step (13) or (14), we need to compute $\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n^{(t)}\|_2^2$ and $\|\mathbf{o}_n^{(t)}\|_2$. Given that $\mathbf{x}_n = \mathbf{X}\mathbf{e}_n$, where $\mathbf{e}_n$ denotes the nth column of $\mathbf{I}_N$, and based on the kernelized updates (28) and (34), it is easy to verify that

$$\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n^{(t)}\|_2^2 = \|\mathbf{X}(\mathbf{e}_n - \boldsymbol{\beta}_c^{(t)} - \boldsymbol{\alpha}_n^{(t)})\|_2^2 \quad (36)$$

$$= \|\mathbf{e}_n - \boldsymbol{\beta}_c^{(t)} - \boldsymbol{\alpha}_n^{(t)}\|_{\mathbf{K}}^2 \quad (37)$$

for every n and c, where $\boldsymbol{\beta}_c^{(t)}$ is the cth column of $\mathbf{B}^{(t)}$. As in (34), it follows that

$$\|\mathbf{o}_n^{(t)}\|_2 = \|\mathbf{X}\boldsymbol{\alpha}_n^{(t)}\|_2 = \|\boldsymbol{\alpha}_n^{(t)}\|_{\mathbf{K}}. \quad (38)$$

The kernelized robust K-means (KRKM) algorithm is summarized as Algorithm 3. As for RKM, the KRKM algorithm is terminated when $\frac{\|\mathbf{M}^{(t)} - \mathbf{M}^{(t-1)}\|_F}{\|\mathbf{M}^{(t)}\|_F} \leq \epsilon_s$, or equivalently due to (28),

$$\frac{\left(\sum_{c=1}^C \|\boldsymbol{\beta}_c^{(t)} - \boldsymbol{\beta}_c^{(t-1)}\|_{\mathbf{K}}^2\right)}{\left(\sum_{c=1}^C \|\boldsymbol{\beta}_c^{(t)}\|_{\mathbf{K}}^2\right)} \leq \epsilon_s^2$$

for a small $\epsilon_s > 0$. KRKM requires $\mathcal{O}(N^3C)$ operations per iteration, whereas the stored variables $\mathbf{A}$, $\mathbf{B}$, $\mathbf{\Delta}$, $\mathbf{U}_q$, and $\mathbf{K}$ occupy $\mathcal{O}(N^2)$ space. Note that if the centroids $\mathbf{M}$ are explicitly needed (e.g., for interpretative purposes), they can be acquired via (28) after KRKM has terminated.

Algorithm 3: Kernelized RKM

Require: Gramian matrix $\mathbf{K} \succ \mathbf{0}$, number of clusters C, $q \geq 1$, and $\lambda > 0$.

1: Initialize $\mathbf{U}^{(0)}$ randomly in $\mathcal{U}_2$, and $\mathbf{A}^{(0)}$ to zero.

2: **for** $t = 1, 2, \ldots$ **do**

3: Update $\mathbf{B}^{(t)}$ from (30).

4: Update $\mathbf{\Delta}^{(t)}$ from (33).

5: Update $\mathbf{A}^{(t)}$ from (35).

6: Update $\mathbf{U}^{(t)}$ and $\mathbf{U}_q^{(t)}$ from (13) or (14), (36), and (38).

7: **end for**

B. Kernelized RKM for Nonlinearly Separable Clusters

Due to the Euclidean distance used, standard K-means tacitly assumes that the underlying clusters are of spherical shape, and linearly separable; GMM-based clustering shares this limitation too. Kernel K-means bypasses this hurdle by mapping vectors $\mathbf{x}_n$ to a higher dimensional space, $\mathcal{H}$, called feature space, through the nonlinear function $\varphi : \mathbb{R}^p \to \mathcal{H}$ [30]. The mapped data $\{\varphi(\mathbf{x}_n)\}_{n=1}^N$ are of dimension $P > p$ or even infinite. K-means in its kernelized version is subsequently applied on $\varphi(\mathbf{x}_n)$. Thus, linearly separable partitions in feature space enable nonlinearly separable partitions in the original data space.

For an algorithm to be kernelizable, the inner products $\varphi^T(\mathbf{x}_n)\varphi(\mathbf{x}_m)$ should be easily computable. When the linear mapping $\varphi(\mathbf{x}_n) = \mathbf{x}_n$ is trivially assumed, these inner products are simply the entries of the Gramian $\mathbf{X}^T\mathbf{X}$. When a nonlinear mapping is used, the so-termed kernel matrix $\mathbf{K}$ with entries $[\mathbf{K}]_{n,m} := \varphi^T(\mathbf{x}_n)\varphi(\mathbf{x}_m)$ replaces the Gramian matrix and must be known. By definition, $\mathbf{K}$ is positive semidefinite and can be employed for (robust) clustering, even when $\varphi(\mathbf{x}_n)$ is high-dimensional (cf. Section IV-A), infinite-dimensional, or even unknown [13]. Of particular interest is the case where $\mathcal{H}$ is a reproducing kernel Hilbert space. Then, the inner product in $\mathcal{H}$ is provided by a known kernel function $\kappa(\mathbf{x}_n, \mathbf{x}_m) := \varphi^T(\mathbf{x}_n)\varphi(\mathbf{x}_m)$ [29, Ch. 3]. Typical kernels for vector data are the polynomial and the Gaussian ones; kernels can be defined for non-vectorial objects too, such as strings or graphs [29].

Building on the KRKM developed in Section IV-A, handling arbitrary kernels is now straightforward. Knowing $\mathbf{X}$ and the kernel $\kappa(\mathbf{x}_n, \mathbf{x}_m)$, matrix $\mathbf{K}$ can be readily computed. By using the kernel in lieu of the Gramian, Algorithm 3 carries over readily to the nonlinear clustering regime. Note however that contrary to clustering high-dimensional data, in (robust) nonlinear clustering centroids cannot be computed in general: even if one is able to recover the feature space centroid, its input space pre-image may not exist [29, Ch. 18].

C. Kernelized Robust Probabilistic Clustering

Kernelizing RPC hinders a major difference over the kernelization of RKM: GMM and RPC updates in Section III-B remain valid for feature vectors only when their dimension P is finite and known. The implication is elucidated as follows. First, updating the variance in (25) entails the underlying dimension p, which becomes P when it comes to kernelization. Second, the (outlier-aware) mixtures of Gaussians degenerate when it comes to modeling infinite-dimensional random vectors. To overcome this limitation, the notion of the empirical kernel map will be exploited [29, Ch. 2.2.6]. Given the input vectors in $\mathcal{X}$ and their kernel matrix $\mathbf{K}$, it is possible to replace φ with the empirical kernel map $\hat{\varphi} : \mathbb{R}^p \to \mathbb{R}^N$ defined as $\hat{\varphi}(\mathbf{x}) := (\mathbf{K}^{\frac{1}{2}})^{\dagger}[\kappa(\mathbf{x}_1, \mathbf{x}) \cdots \kappa(\mathbf{x}_N, \mathbf{x})]^T$, where $(\cdot)^{\dagger}$ denotes the Moore–Penrose pseudoinverse. The feature space $\hat{\mathcal{H}}$ induced by $\hat{\varphi}$ has finite dimensionality N, while it can be verified that $\hat{\varphi}^T(\mathbf{x}_n)\hat{\varphi}(\mathbf{x}_m) = \varphi^T(\mathbf{x}_n)\varphi(\mathbf{x}_m) = \kappa(\mathbf{x}_n, \mathbf{x}_m)$ for all $\mathbf{x}_n, \mathbf{x}_m \in \mathcal{X}$.

In the kernelized probabilistic setup, $\hat{\varphi}(\mathbf{x}_n)$'s are assumed drawn from a mixture of C multivariate Gaussian distributions with $\mathbf{\Sigma} = \sigma^2 \boldsymbol{I}_N$ common to all clusters. The EM-based updates of RPC in Section III-B remain valid after replacing the dimension p in (25) by N, and the input vectors $\mathbf{x}_n$'s by $\hat{\varphi}(\mathbf{x}_n)$'s whose inner products are the entries of $\mathbf{K}$. The kernelization procedure is similar to the one followed for RKM: first, the auxiliary matrices $\mathbf{A}^{(t)}, \mathbf{B}^{(t)}$, and $\mathbf{\Delta}^{(t)}$ are introduced. By randomly initializing with $\sigma^{(0)}$, $\boldsymbol{\pi}^{(0)} \in \mathcal{P}$, $\mathbf{B}^{(0)} \in \mathbb{R}^{N \times C}$, and setting $\mathbf{A}^{(0)}$ to zero, it can be shown as in Proposition 4, that the kernelized RPC updates for $\mathbf{O}^{(t)}$, $\mathbf{M}^{(t)}$, and $\mathbf{R}^{(t)}$ have their columns lying in $\mathrm{range}(\mathbf{\Phi})$, where $\mathbf{\Phi} := [\hat{\varphi}(\mathbf{x}_1) \cdots \hat{\varphi}(\mathbf{x}_N)]$. Instead of the assignment matrix $\mathbf{U}$ in KRKM, the $N \times C$ matrix of posterior probability estimates $\mathbf{\Gamma}^{(t)}$ is used, where $[\mathbf{\Gamma}^{(t)}]_{n,c} := \gamma_{nc}^{(t)}$ satisfying $\mathbf{\Gamma}^{(t)}\mathbf{1}_C = \mathbf{1}_N \; \forall t$. The kernelized RPC (KRPC) algorithm is summarized as Algorithm 4. As with KRKM, its computations are $\mathcal{O}(N^3C)$ per iteration, whereas the stored variables $\mathbf{A}$, $\mathbf{B}$, $\mathbf{\Delta}$, $\mathbf{\Gamma}$, $\boldsymbol{\pi}$, $\mathbf{K}$, and σ occupy $\mathcal{O}(N^2)$ space.

Algorithm 4: Kernelized RPC

Require: Gramian or kernel matrix $\mathbf{K} \succ \mathbf{0}$, number of clusters C, and $\lambda > 0$.

1: Randomly initialize $\sigma^{(0)}$, $\boldsymbol{\pi}^{(0)} \in \mathcal{P}$, and $\mathbf{B}^{(0)}$; and set $\mathbf{A}^{(0)}$ to zero.

2: **for** $t = 1, 2, \ldots$ **do**

3: Update $\mathbf{\Gamma}^{(t)}$ via (18) exploiting (36).

4: Update $\boldsymbol{\pi}^{(t)}$ as $\boldsymbol{\pi}^{(t)} = (\mathbf{\Gamma}^{(t)})^T \frac{\mathbf{1}_N}{N}$.

5: Update $\mathbf{B}^{(t)}$ as $\mathbf{B}^{(t)} = (\mathbf{I}_N - \mathbf{A}^{(t-1)})\mathbf{\Gamma}^{(t)}\mathrm{diag}^{-1}(N\boldsymbol{\pi}^{(t)})$.

6: Update $\mathbf{\Delta}^{(t)}$ as $\mathbf{\Delta}^{(t)} = \mathbf{I}_N - \mathbf{B}^{(t)}(\mathbf{\Gamma}^{(t)})^T$.

7: Update the columns of $\mathbf{A}^{(t)}$ as $\boldsymbol{\alpha}_n^{(t)} = \boldsymbol{\delta}_n^{(t)} \left[1 - \frac{\lambda\sigma^{(t-1)}}{\|\boldsymbol{\delta}_n^{(t)}\|_{\mathbf{K}}}\right]_+$ for all n.

8: Update $\sigma^{(t)}$ via (25) where p is replaced by N, using the ℓ_2-norms computed in Step 3, and exploiting $\|\mathbf{o}_n^{(t)}\|_2 = \|\boldsymbol{\alpha}_n^{(t)}\|_{\mathbf{K}}$ for all n.

9: **end for**

Remark 5 (Reweighted Kernelized Algorithms): As in Section III-C, reweighted versions of KRKM and KRPC can be derived simply by introducing an iteration-dependent parameter $\lambda_n^{(t)} = \frac{\lambda}{(\|\mathbf{o}_n^{(t-1)}\|_2 + \epsilon)}$.

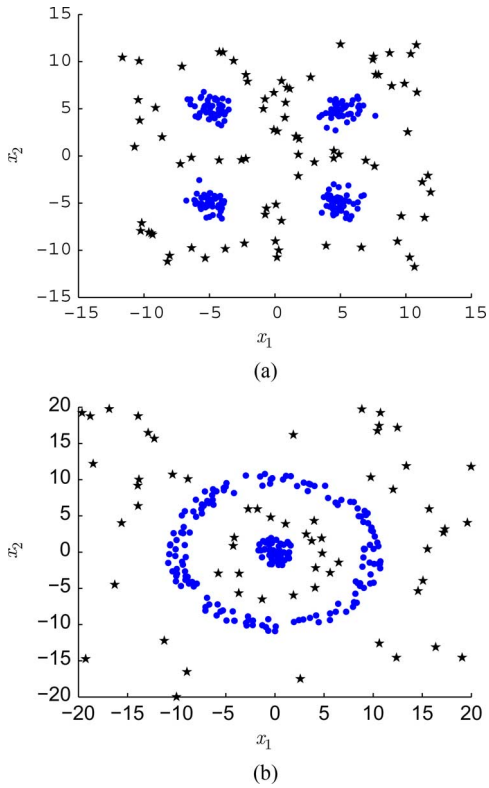


Fig. 1. Synthetic datasets: (Non-)outlier vectors are denoted by circles • (stars ★). (a) Dataset with $C = 4$ spherical clusters and 80 outliers. (b) Dataset with $C = 2$ concentric rings and 60 outliers.

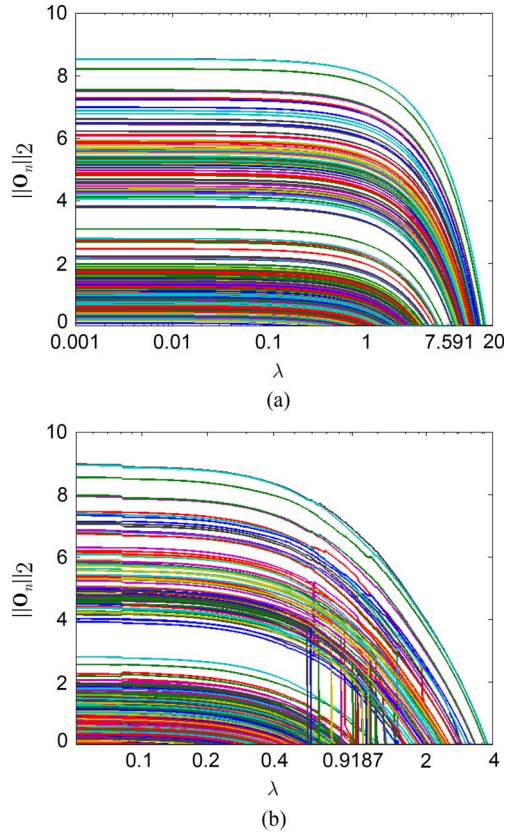


Fig. 2. Curves of $\|\mathbf{o}_n\|_2$'s as a function of λ for the dataset in Fig. 1(a). (a) RKM algorithm for $q = 1$. (b) RPC algorithm.

V. Numerical Tests

Numerical tests illustrating the performance of the novel algorithms on both synthetic and real datasets are presented in this section. Performance is assessed through their ability to identify outliers and the quality of clustering itself. The latter is measured via the adjusted rand index (ARI) between the resulting clustering and the true labels of the data [21]. For methods unable to identify outliers, the ARI is inevitably computed over all data. For methods with outlier detection capabilities, the ARI is computed after excluding the outliers. In each experiment, λ is tuned using the grid search outlined in Remark 4. Thanks to the warm-start technique, the solution path for all grid points is computed in an amount of time comparable to that used for solving for a specific value of λ.

A. Synthetic Datasets

Two synthetic datasets were used. The first one, shown in Fig. 1(a), consisted of a random draw of 200 vectors from $C = 4$ bivariate Gaussian distributions (50 vectors per distribution), and 80 outlying vectors $(N = 280)$. The Gaussian distributions have different means and a common covariance matrix $0.8 \cdot \mathbf{I}_2$. The second dataset comprised points belonging to $C = 2$ concentric rings as depicted in Fig. 1(b). The inner (outer) ring had 50 (150) points. It also contained 60 points lying in-between the rings and outside the outer ring corresponding to outliers $(N = 260)$. Clustering this second dataset is challenging even if outliers were not present due to the shape and multiscale nature of the clusters.

The effect of λ on the number of outliers identified was investigated for the dataset with spherical clusters. In Fig. 2, the values of $\{\|\mathbf{o}_n\|_2\}_{n=1}^N$ are plotted as a function of λ (cf. Remark 4). The outlier-norm curves shown in Fig. 2(a) correspond to the RKM algorithm with $q = 1$ using a random initialization. For $\lambda > 17$, all $\mathbf{o}_n$'s were set to zero. As λ approached zero more $\mathbf{o}_n$'s took nonzero values. Selecting $\lambda \in [6.2, 7.6]$ yielded 80 outliers. Fig. 2(b) shows $\{\|\mathbf{o}_n\|_2\}_{n=1}^N$ as λ varies for the RPC algorithm assuming $\mathbf{\Sigma} = \sigma^2 \boldsymbol{I}_p$. Note that the paths followed by some $\|\mathbf{o}_n\|_2$'s as λ decreases exhibit a fast transition from zero. Focusing on these points, it was empirically observed that when they had zero $\mathbf{o}_n$'s, their posteriors γ_{nc}'s were ambiguous for membership assignment. Upon decreasing λ so that $\|\mathbf{o}_n\|_2 > 0$, one of their γ_{nc}'s quickly becomes unity while the other ones quickly drop to zero, hence, causing $\|\mathbf{o}_n\|_2$ to rapidly increase to some finite value (cf. (23)). It is worth mentioning that this behavior does not entail instability or artifacts in identifying outliers.

In Fig. 3, the number of points identified as outliers, i.e., the number of nonzero $\|\mathbf{o}_n\|_2$'s, is plotted as a function of λ. These curves are useful when setting the value of λ to identify a prescribed number of outliers. The goal here was to identify $s = 80$ outliers. Both RKM and RPC, with λ tuned to detect 80 outliers, were able to correctly cluster the data and identify the outliers. Although obtaining the curves in Fig. 3 entails solving several robust clustering problems, one for each value of λ considered, they can be computed efficiently using warm starts as described in Remark 4.

For this experiment, Fig. 3 also suggested estimating the number of outliers in the dataset by inspecting the curve slopes. When decreasing λ for hard RKM, a plateau results

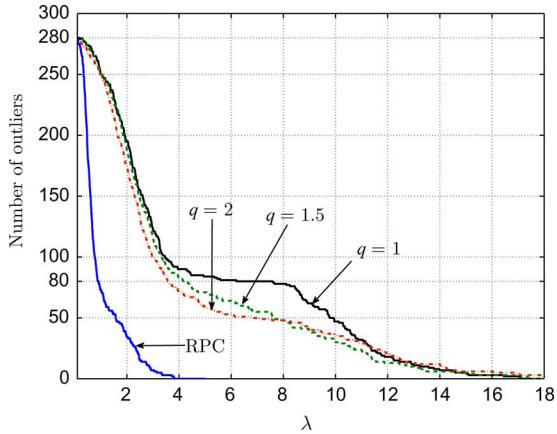


Fig. 3. Number of outliers identified as a function of λ for the dataset in Fig. 1(a).

for $\lambda \in [6.2, 7.6]$, followed by a region with increased slope. This plateau defined a transition region between correctly identifying outliers and erroneously deeming non-outliers as outliers. Similar curve slope changes were observed for soft RKM and RPC around the values of the λ's yielding the correct number of 80 outlying points.

The root-mean-squared error (RMSE) between the cluster centroids estimated by the clustering methods and the sample mean for each cluster was used as a figure of merit. Table I shows the minimum RMSE obtained over 100 random initializations for several values of outlier contamination. All tested algorithms shared common initializations. The tested algorithms were weighted RKM (WRKM) and weighted RPC (WRPC); hard K-means; soft K-means with $q = 1.5$; EM; noise clustering (NC) [10]; α-cut [36]; and possibilistic clustering [24]. The noise distance in NC was chosen so that the noise cluster had the prescribed number of outliers and the tuning parameters for possibilistic clustering were set to 2 for all clusters. RKM and RPC achieved lower RMSE than their non-robust counterparts and α-cut, and were able to correctly identify the outliers in all cases. Noticeable improvement was achieved by WRKM and WRPC, which exhibited the best performance among all algorithms tested. Note that the ARI of the novel robust clustering algorithms corresponding to the RMSE values in Table I was one. Surprisingly, the heuristic NC offered competitive clustering performance after carefully tuning its parameter. Although possibilistic clustering also offers competitive performance, it was empirically observed that it is sensitive to initialization and parameter tuning.

Next, the dataset with concentric circles shown in Fig. 1(b) was clustered using the Gaussian kernel $\kappa(\mathbf{x}_n, \mathbf{x}_m) = \exp(-\alpha_\kappa \|\mathbf{x}_n - \mathbf{x}_m\|_2^2)$, where $\alpha_\kappa > 0$ is a scaling parameter. The parameter α_κ^{-1} was chosen as a robust variance estimate of the entire dataset as described in [8]. Both KRKM and KRPC were able to identify the 60 outlying points. In Fig. 4, the number of outliers identified by KRKM and KRPC is plotted as a function of λ for different values of α_κ. Fig. 5 illustrates the values of $\|\mathbf{o}_n\|_2$'s for WKRKM and WKRPC when seeking 60 outliers. Points surrounded by a circle correspond to vectors identified as outliers, and each circle's radius is proportional to its corresponding $\|\mathbf{o}_n\|_2$ value.

TABLE I
RMSE PERFORMANCE OF CLUSTERING ALGORITHMS FOR DATASET WITH $C = 4$ SPHERICAL CLUSTERS

	RMSE				
Outliers/N	**10**/210	**20**/220	**40**/240	**60**/260	**80**/280
hard K-means	0.6002	0.7713	1.2009	1.3927	1.5856
soft K-means ($q = 1.5$)	0.5156	0.6546	1.2006	1.5014	1.4558
EM	0.6003	0.7607	1.1267	1.2961	1.5271
hard RKM	0.2505	0.3660	0.6242	0.800	1.0126
hard WRKM	**0.0710**	**0.0627**	**0.0739**	**0.0461**	**0.0723**
soft RKM ($q = 1.5$)	0.2162	0.2129	0.3170	0.3706	0.4981
soft WRKM ($q = 1.5$)	**0.0521**	**0.0389**	**0.0304**	**0.0359**	**0.0407**
RPC	0.2984	0.3393	0.4483	0.5597	0.6652
WRPC	**0.0366**	**0.0572**	**0.0019**	**0.0029**	**0.0615**
α-cut ($\alpha = 0.5$)	0.6002	0.7713	1.2070	1.4264	1.6646
α-cut ($\alpha = 0.7$)	0.6078	0.7754	1.1671	1.3603	1.6911
α-cut ($\alpha = 0.9$)	0.5375	0.7006	1.1196	1.3088	1.6190
soft NC ($q = 1.5$)	0.0479	0.0598	0.0526	0.0493	0.0696
possibilistic ($q = 1.5$)	0.3207	0.3208	0.3207	0.3202	0.3201

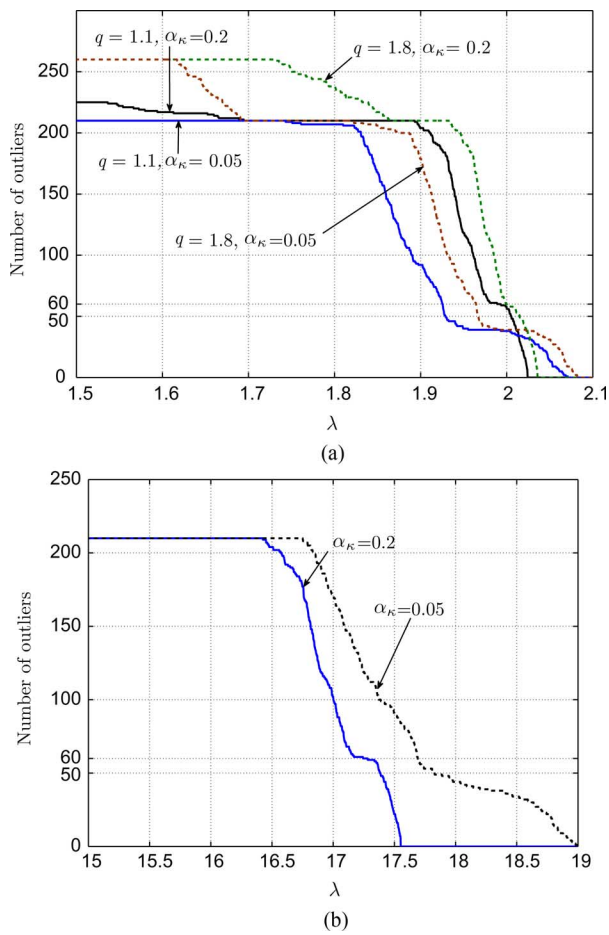


Fig. 4. Number of outliers identified as a function of λ for the dataset in Fig. 1(b). (a) KRKM algorithm. (b) KRPC algorithm.

B. USPS Dataset

In this subsection, the robust clustering algorithms were tested on the United States Postal Service (USPS) handwritten digit recognition corpus. This corpus contained gray-scale digit images of 16×16 pixels with intensities normalized to $[-1, 1]$.

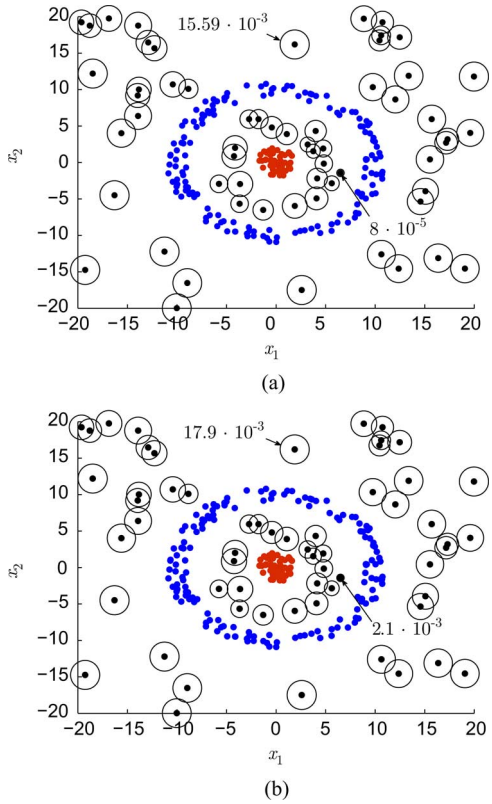


Fig. 5. Clustering results for the dataset in Fig. 1(b) using a Gaussian kernel with $\alpha_\kappa = 0.2$. Points surrounded by a circle were deemed as outliers; the radius of the circle is proportional to the value of $\|\mathbf{o}_n\|_2$. Smallest and largest $\|\mathbf{o}_n\|_2$ values are shown. (a) KRKM algorithm ($q = 1.1$). (b) KRPC algorithm.

TABLE II
ARI FOR THE USPS DATASET ($C = 6$)

Kernel	**K-means**	**K-medians**	**RKM**	**RPC**
Linear	0.6469	0.5382	0.6573	0.6508
Polynomial	0.5571	-	0.6978	0.6965

It was divided into 7201 training and 2007 test examples of the digits 0–9. Although the corpus contained class labels, they were known to be inconsistent: some digits were erroneously labeled, while some images were difficult to be classified even by humans [29, App. A]. In this experiment, the subset of digits 0–5 was used. For each digit, both training and test sets were combined to a single set and then 300 images were sampled uniformly at random, yielding a dataset of 1800 images. Each image was represented by a 256-dimensional vector normalized to have unit ℓ_2-norm.

RKM ($q = 1$) and RPC were used to partition the dataset into $C = 6$ clusters and identify $s = 100$ outliers. A total of 20 Monte Carlo runs with random initializations common to all algorithms were performed. The final clustering was chosen as the one attaining the smallest cost in (5). The ARI's for K-means, K-medians, and the proposed schemes are shown in Table II. Both RKM and RPC show improved clustering performance than the non-robust algorithms. Also, the ARI obtained by WRKM (WRPC) was equal to the one obtained by RKM (RPC). Note that, the K-medians algorithm was unable to find a partitioning for the data revealing the 6 digits present even after 100 Monte Carlo runs.

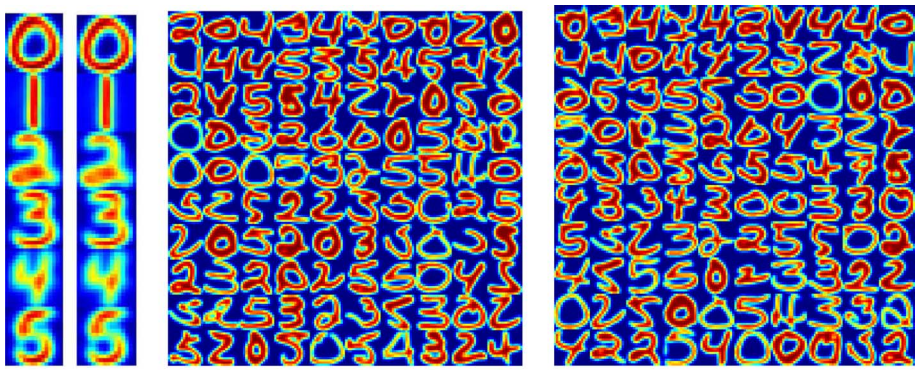
The USPS dataset was clustered using the RKM and WRKM tuned to identify 100 outliers. WRKM was initialized with the results obtained by RKM. Although RKM and WRKM yielded the same outlier images, the size of the $\mathbf{o}_n$'s was different, becoming nearly uniform for WRKM. The USPS dataset was also clustered using the RPC and the WRPC algorithms. Fig. 6(a) shows the cluster centroids obtained by RPC and WRPC. Fig. 6(b) shows the 100 outliers identified. The outliers identified by the RPC and WRPC algorithms also coincide. The position of the outlier images in the mosaic corresponds to their ranking according to the size of their corresponding $\mathbf{o}_n$ (largest to smallest from left to right, top to bottom). Note that all outliers identified have a trait that differentiates them from the average image in each cluster. Among the 100 outliers detected by RKM and RPC, 97 were common to both approaches.

Kernelized versions of the algorithms were also used on the USPS dataset. Similar to [29], the homogeneous polynomial kernel of order 3, that is $\kappa(\mathbf{x}_n, \mathbf{x}_m) = (\mathbf{x}_n^T \mathbf{x}_m)^3$, was used. The ARI scores obtained by the kernelized robust clustering algorithms are shown in Table II. Based on these scores, two important observations are in order: i) kernelized K-means is more sensitive to outliers than K-means; but ii) KRKM for the particular kernel yields an improved clustering performance over RKM. Finally, the 100 outliers identified by KRKM are shown in Fig. 6(c).

C. Document Clustering

The KRKM algorithm developed in Section IV-A was evaluated next. The context here is document clustering on the standard TDT2 dataset. The TDT2 corpus consists of data collected during the first half of 1998 from six news sources [6]. It comprises documents classified into 96 semantic categories. Every document is represented by a vector containing the times each one of $p = 36\,771$ terms occurs in the document. After discarding the documents assigned to more than one category, one ends up with 10 212 document vectors, which are subsequently normalized to have unit ℓ_2-norm. The categories range from 1 to 1844 documents. For each clustering experiment, 4 out of the 19 largest categories are randomly selected, and 100 documents from each of these 4 categories are uniformly sampled. An additional random sample of 20 documents from the smallest 30 categories comprises the additive outliers, yielding a total of $N = 420$ documents. By hiding the category labels from the algorithms, the task here is to jointly cluster the 400 documents into 4 large categories and identify the 20 ones drawn from the smaller categories as outliers. The Gaussian kernel with unit bandwidth is utilized, i.e., $[\mathbf{K}]_{n,m} = \exp(-\|\mathbf{x}_n - \mathbf{x}_m\|_2^2)$, and documents are partitioned into $C = 4$ clusters [6].

The hard KRKM (cf. Alg. 3) is compared against NC and α-cut [10], [36]. All algorithms tested are initialized to the partitioning found as follows: the $N \times C$ matrix containing the eigenvectors corresponding to the C largest eigenvalues of $\mathbf{K}$ is input to the standard K-means algorithm, which assigns its N rows to C classes [12]. The figure of merit here is detectability of outlying documents. For each sample dataset, all three algorithms are run for a grid of values for their tuning parameters (λ for KRKM, the distance to the noise cluster in NC, and the percentage α for the α-cut method). Hence, for each dataset, pairs of false alarm and correct detection probabilities are obtained,



(a) (b) (c)

Fig. 6. Clustering and outliers for the USPS dataset with $C = 6$ tuned to identify $s = 100$ outliers (a) RPC and WRPC centroids (b) Outliers identified by RPC and WRPC (c) Outliers identified by KRKM using the polynomial kernel of order 3.

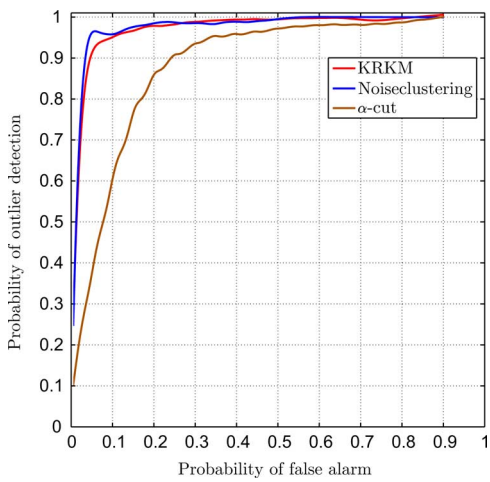


Fig. 7. Identification of outlying documents in TDT2 corpus.

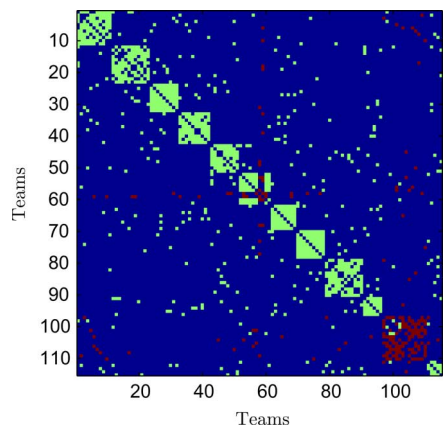


Fig. 8. The kernel matrix for the college football network permuted using KRKM clustering. Zero entries are colored blue and outliers are colored red.

which correspond to points of the receiver operating characteristic (ROC) curve. ROC curves of 50 sample datasets are averaged via smoothing splines with parameter 0.9999, and the results are plotted in Fig. 7. KRKM exhibits detection performance comparable to NC, while they both outperform the α-cut method.

D. College Football Network

KRKM was used to partition and identify outliers in a network of $N = 115$ college football teams playing in 12 different conferences for Division I games during year 2000 [17]. In this schedule, teams played more often against teams in the same conference. Each node in the network corresponds to a team and a link between two teams exists if they played against each other during the season. The network structure is summarized by the $N \times N$ adjacency matrix $\mathbf{E}$.

To identify groups and outliers, the connection between kernel K-means and spectral clustering for graph partitioning was exploited [12]. According to this connection, the conventional spectral clustering algorithm is substituted by kernelized K-means using a specific kernel matrix. The kernel matrix used was $\mathbf{K} = \nu \boldsymbol{I}_N + \mathbf{D}^{\frac{-1}{2}}\mathbf{E}\mathbf{D}^{\frac{-1}{2}}$, where $\mathbf{D} := \mathrm{diag}(\mathbf{E}\mathbf{1}_N)$ and ν was chosen such that $\mathbf{K} \succ \mathbf{0}$. The teams were divided into $C = 12$ groups. KRKM was initialized via spectral clustering and λ was tuned to identify $s = 12$ outliers. Fig. 8 shows the entries of the kernel matrix $\mathbf{K}$ after being row and column permuted so that teams in the same cluster appear grouped. The ARI obtained by KRKM was 0.9218.

Teams identified as outliers sorted in descending order based on their $\|\mathbf{o}_n\|_2$ values are: Connecticut, Navy, Notre Dame, Northern Illinois, Toledo, Miami (Ohio), Bowling Green State, Central Michigan, Eastern Michigan, Kent, Ohio, and Marshall. Three of them, namely Connecticut, Notre Dame, and Navy, were independent teams. Connecticut was assigned to the Mid-American conference, but it did not play as many games with teams from this conference (4 games) as other teams in the same conference did (around 8 games). Notre Dame and Navy played an equal number of games with teams from two different conferences so they could be assigned to either one. Several teams from the Mid-American conference were categorized as outliers. In hindsight, this can be explained by the subdivision of the conference into East and West Mid-American conferences. Teams in each of the Mid-American sub-conferences played about the same number of games with teams from their own sub-conference and the rest of the teams. Interestingly, the sub-partition of the Mid-American conference was identified by using KRKM with $C = 13$ while still seeking for 12 outliers. In this case, the ARI for the partition was 0.9110. The three independent teams, Connecticut, Notre Dame, and Navy, were again among the 12 outliers identified.

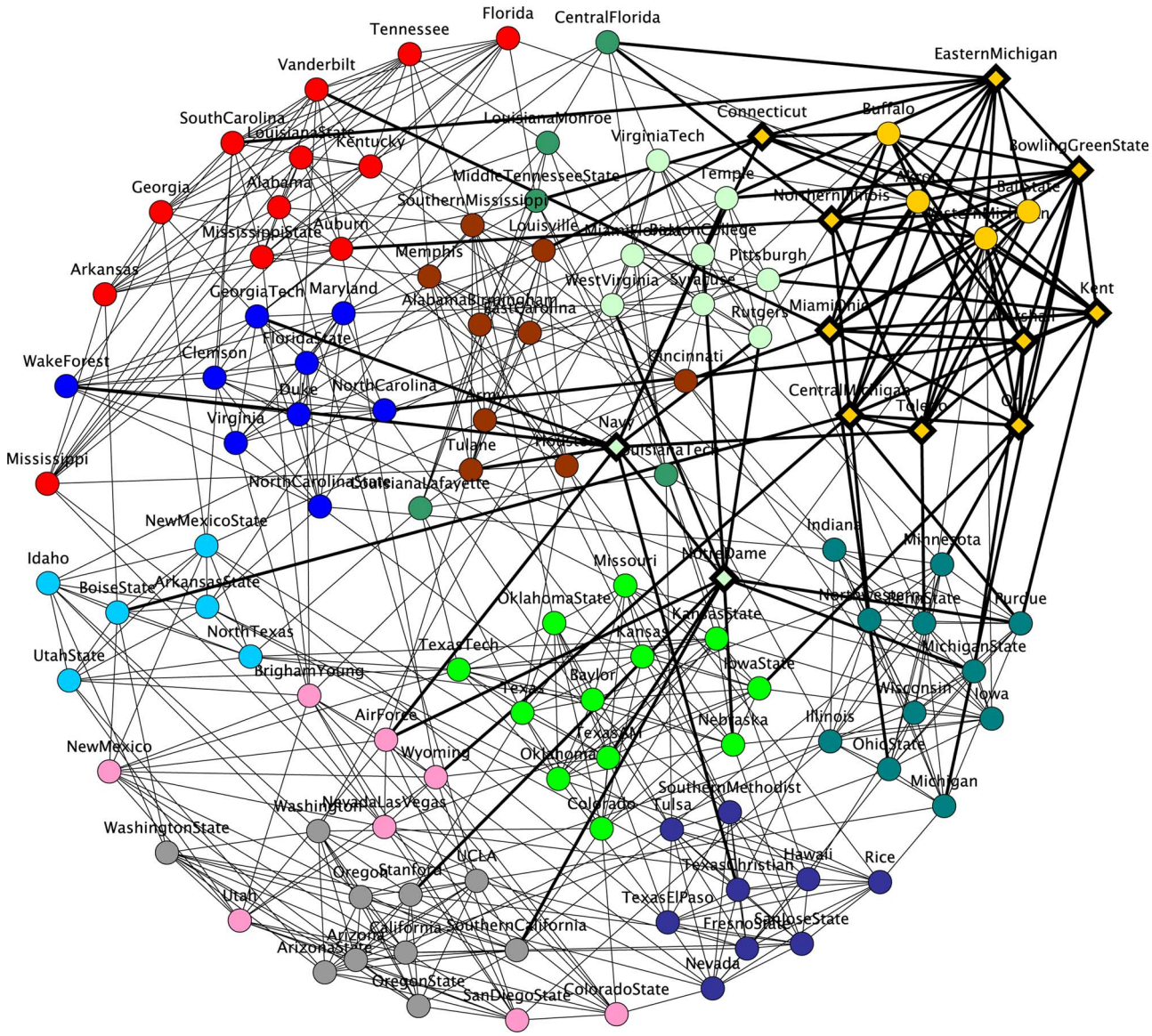


Fig. 9. Clustering of the college football network obtained by KRKM for $C = 12$. Outliers are represented by diamond-shaped nodes.

VI. Conclusion

Algorithms were developed for robust clustering based on a principled data model accounting for outliers. Both deterministic and probabilistic partitional clustering setups based on the K-means and GMM-based algorithms were considered. Exploiting the fact that outliers appear infrequently in the data, a neat connection with sparsity-aware signal processing algorithms was made. This led to the development of computationally efficient and provably convergent robust clustering algorithms. Kernelized versions of the algorithms, well-suited for high-dimensional data or when only similarity information among objects is available, were also developed. The performance of the robust clustering algorithms was validated via numerical experiments both on synthetic and real datasets.

Appendix A
Proof of Proposition 1

Since $\sum_{c=1}^{C}(u_{nc}^{(t)})^q > 0$ for all n and t due to (c3), the first summand of $\phi^{(t)}(\mathbf{o}_n)$ in (10) is a strictly convex function of $\mathbf{o}_n$. Hence, $\phi^{(t)}(\mathbf{o}_n)$ is a strictly convex function too and its minimizer is unique. Then, recall that a vector $\mathbf{o}_n^{(t)}$ is a minimizer of (10) if and only if $\mathbf{0} \in \partial\phi^{(t)}(\mathbf{o}_n^{(t)})$, where $\partial\phi^{(t)}(\mathbf{o}_n)$ is the sub-differential of $\phi^{(t)}(\mathbf{o}_n)$. For $\mathbf{o}_n \neq \mathbf{0}$, where the cost in (10) is differentiable, $\partial\phi^{(t)}(\mathbf{o}_n)$ is simply the gradient $-2\sum_{c=1}^{C}(u_{nc}^{(t-1)})^q\left(\mathbf{x}_n - \mathbf{m}_c - \left(1+\frac{\lambda}{2\|\mathbf{o}_n\|_2}\right)\mathbf{o}_n\right)$. At $\mathbf{o}_n = \mathbf{0}$, the sub-differential of the ℓ_2-norm $\|\mathbf{o}_n\|_2$ is the set of vectors $\{\mathbf{v}_n : \|\mathbf{v}_n\|_2 \leq 1\}$ by definition, and then the sub-differential of $\phi^{(t)}(\mathbf{o}_n)$ is $\partial\phi^{(t)}(\mathbf{o}_n) = \left\{-2\sum_{c=1}^{C}(u_{nc}^{(t-1)})^q \left(\mathbf{x}_n - \mathbf{m}_c - \frac{\lambda}{2}\mathbf{v}_n\right) : \|\mathbf{v}_n\|_2 \leq 1\right\}$.

When the minimizer $\mathbf{o}_n^{(t)}$ is nonzero, the condition $\mathbf{0} \in \partial\phi^{(t)}(\mathbf{o}_n^{(t)})$ implies

$$\left(1 + \frac{\lambda}{2\|\mathbf{o}_n^{(t)}\|_2}\right)\mathbf{o}_n^{(t)} = \mathbf{r}_n^{(t)} \tag{A.39}$$

where $\mathbf{r}_n^{(t)}$ has been defined in (12). Equation (A.39) reveals that $\mathbf{o}_n^{(t)}$ is a positively scaled version of $\mathbf{r}_n^{(t)}$. The scaling can be readily found by taking the ℓ_2-norm on both sides of (A.39),

i.e., $\|\mathbf{o}_n^{(t)}\|_2 = \|\mathbf{r}_n^{(t)}\|_2 - \frac{\lambda}{2}$, which is valid for $\|\mathbf{r}_n^{(t)}\|_2 > \frac{\lambda}{2}$. Substituting this back to (A.39), yields $\mathbf{o}_n^{(t)} = \mathbf{r}_n^{(t)} \left(1 - \frac{\lambda}{2\|\mathbf{r}_n^{(t)}\|_2}\right)$.

For $\mathbf{o}_n^{(t)} = \mathbf{0}$, there exists a $\mathbf{v}_n^{(t)}$ for which $\|\mathbf{v}_n^{(t)}\|_2 \leq 1$ and $\mathbf{v}_n^{(t)} = \left(\frac{2}{\lambda}\right)\mathbf{r}_n^{(t)}$. This is possible when $\|\mathbf{r}_n^{(t)}\|_2 \leq \frac{\lambda}{2}$. These two cases for $\mathbf{o}_n^{(t)}$ are compactly expressed via (11).

Appendix B
Proof of Proposition 2

By defining $f_s(c)$ as being zero when the Boolean argument c is true, and ∞ otherwise, the problem in (6) can be written in the unconstrained form

$$\min_{\mathbf{M},\mathbf{O},\mathbf{U}} \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}^{q}\left(\|\mathbf{x}_n - \mathbf{m}_c - \mathbf{o}_n\|_2^2 + \lambda\|\mathbf{o}_n\|_2\right) + f_s(\mathbf{U} \in \mathcal{U}_2). \quad \text{(B.40)}$$

The cost in (B.40), call it $f(\mathbf{M},\mathbf{O},\mathbf{U})$, is a proper and lower semi-continuous function, which implies that its non-empty level sets are closed. Also, since f is coercive, its level sets are bounded. Hence, the non-empty level sets of f are compact. For $q > 1$, function $f(\mathbf{M},\mathbf{O},\mathbf{U})$ has a unique minimizer per optimization block variable $\mathbf{M}$, $\mathbf{O}$, and $\mathbf{U}$. Then, convergence of the RKM algorithm to a coordinate-wise minimum point of (6) follows from [33, Th. 4.1(c)].

When $q = 1$, define the first summand in (B.40) as $f_0(\mathbf{M},\mathbf{O},\mathbf{U}) := \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}\|\mathbf{x}_n - \mathbf{m}_c - \mathbf{o}_n\|_2^2$, which is the differentiable part of f. Function f_0 has an open domain, and the remaining non-differentiable part of f is separable with respect to the optimization blocks. Hence, again by [33, Th. 4.1(c)], the RKM algorithm with $q = 1$ converges to a local minimum $(\mathbf{M}^*,\mathbf{O}^*,\mathbf{U}^*)$ of (6).

It has been shown so far that for $q = 1$, a BCD iteration converges to a local minimum of (6). The BCD step for updating $\mathbf{U}$ is the hard rule in (14). Hence, this BCD algorithm i) yields a $\mathbf{U}^*$ with binary entries, and ii) essentially implements the BCD updates for solving (5). Since a local minimum of (6) with binary assignments is also a local minimum of (5), the claim of the proposition follows.

Appendix C
Proof of Proposition 3

Combining the two steps of the EM algorithm, namely (18) and (19), it is easy to verify that the algorithm is equivalent to a sequence of BCD iterations for optimizing

$$\min_{\mathbf{\Gamma},\mathbf{\Theta}'} -\sum_{n=1}^{N}\sum_{c=1}^{C}\gamma_{nc}\log\left(\frac{\pi_c\mathcal{N}(\mathbf{x}_n;\mathbf{m}_c+\mathbf{o}_n,\mathbf{\Sigma})}{\gamma_{nc}}\right) + \lambda\sum_{n=1}^{N}\|\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}} + f_s(\mathbf{\Gamma}\in\mathcal{U}_2) + f_s(\boldsymbol{\pi}\in\mathcal{P}) + f_s(\mathbf{\Sigma}\succ\mathbf{0}) \quad \text{(C.41)}$$

where $\mathbf{\Theta}' := \{\boldsymbol{\pi},\mathbf{M},\mathbf{O},\mathbf{\Sigma}\}$, the $N \times C$ matrix $\mathbf{\Gamma}$ has entries $[\mathbf{\Gamma}]_{n,c} := \gamma_{nc} > 0$, and as in (B.40) that $f_s(c)$ is zero when condition c is true, and ∞ otherwise. That the $\{\gamma_{nc}\}$ are positive follows after using Bayes' rule to deduce that $\gamma_{nc} \propto \pi_c\mathcal{N}(\mathbf{x}_n;\mathbf{m}_c+\mathbf{o}_n,\mathbf{\Sigma})$ and noticing that (i) $\mathcal{N}(\mathbf{x}_n;\mathbf{m}_c+\mathbf{o}_n,\mathbf{\Sigma})$ is positive for all $\mathbf{x}_n$, and (ii) all π_c must be positive so that the cost in (C.41) remains finite.

The objective function of this minimization problem is proper, bounded below, and lower semi-continuous implying, that its non-empty level sets are closed. Since this function is also coercive, its level sets are bounded. Hence, its non-empty level sets are compact. Moreover, the objective function has a unique minimizer for the optimization blocks $\boldsymbol{\pi}$, $\mathbf{M}$, and $\mathbf{O}$. In particular, the $\mathbf{M}$ block minimizer is unique since $\sum_{n=1}^{N}\gamma_{nc} > 0$, for all $c \in \mathbb{N}_C$. Then, by [33, Th. 4.1 (c)], the RPC algorithm converges to a coordinate-wise minimum point of (7).

References

[1] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *J. Mach. Learn. Res.*, vol. 2, pp. 125–137, Dec. 2001.

[2] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Norwell, MA: Kluwer, 1981.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[4] L. Bobrowski and J. C. Bezdek, "C-means clustering with the l_1 and l_∞ norms," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 545–554, May 1991.

[5] H.-H. Bock, "Clustering methods: A history of K-means algorithms," in *Selected Contributions in Data Analysis and Classification*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, P. Brito, G. Cucumel, P. Bertr, and F. Carvalho, Eds. Berlin, Germany: Springer, 2007, pp. 161–172.

[6] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[7] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[8] Y. Chen, X. Dang, H. Peng, and H. L. Bart, "Outlier detection with the kernelized spatial depth function," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 288–305, Feb. 2009.

[9] S. Dasgupta and Y. Freund, "Random projection trees for vector quantization," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3229–3242, Jul. 2009.

[10] R. N. Davé and R. Krishnapuram, "Robust clustering methods: A unified view," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 270–293, 1997.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., Series B (Methodol.)*, vol. 39, pp. 1–38, Aug. 1977.

[12] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, Seattle, WA, 2004, pp. 551–556.

[13] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: A multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1944–1957, Nov. 2007.

[14] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, "Pathwise coordinate optimization," *Ann. Appl. Stat.*, vol. 1, no. 2, pp. 302–332, 2007.

[15] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 450–465, May 1999.

[16] L. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar, "A review of robust clustering methods," *Adv. Data Anal. Classificat.*, vol. 4, no. 2, pp. 89–109, 2010.

[17] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. of Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer Series in Statistics. New York: Springer, 2009.

[19] K. Honda, A. Notsu, and H. Ichihashi, "Fuzzy PCA-guided robust K-means clustering," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, pp. 67–79, Feb. 2010.

[20] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. New York: Wiley, 2009.

[21] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.
[22] J. M. Jolion, P. Meer, and S. Bataouche, "Robust clustering with applications in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 791–802, Aug. 1991.
[23] P. R. Kersten, "Fuzzy order statistics and their application to fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 708–712, Dec. 1999.
[24] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, May 1993.
[25] K. Lange, D. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions (with discussion)," *J. Comput. Graphic. Stat.*, vol. 9, pp. 1–59, 2000.
[26] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
[27] D. T. Nguyen, L. Chen, and C. K. Chan, "An outlier-aware data clustering algorithm in mixture models," in *Proc. 7th Intl. Conf. Inf., Commun,, Signal Process.*, Macau, China, Dec. 2009, pp. 1–5.
[28] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy C-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, Aug. 2005.
[29] B. Schölkopf and A. J. Smola*, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
[30] B. Schölkopf, A. J. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[31] D. W. Scott, "Outlier detection and clustering by partial mixture modeling," in *Proceedings in Computational Statistics*, J. E. Antoch, Ed. Prague, Chezh Republic: Physica-Verlag, 2004, pp. 453–465.
[32] S. Z. Selim and M. A. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 1, pp. 81–86, Jan. 1984.
[33] P. Tseng, "Convergence of block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.
[34] J. Weston, A. Elisseff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *J. Mach. Learn. Res.*, vol. 3, pp. 1439–1461, Mar. 2003.
[35] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.
[36] M. S. Yang, K. L. Wu, J. N. Hsieh, and J. Yu, "Alpha-cut implemented fuzzy clustering algorithms and switching regressions," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, pp. 588–603, 2008.
[37] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Stat. Soc., Series B*, vol. 68, no. 1, pp. 49–67, Feb. 2006.
[38] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao, "Gaussian mixture density modeling, decomposition, and applications," *IEEE Trans. Image Process.*, vol. 5, no. 9, pp. 1293–1302, Sep. 1996.

Pedro A. Forero (S'07) received the Diploma degree in electronics engineering from Pontificia Universidad Javeriana, Bogota, Colombia, in 2003 and the M.Sc. degree in electrical engineering from Loyola Marymount University in 2006.

Since September 2006, he has been working towards the Ph.D. degree with the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. His research interests include statistical signal processing, machine learning and network science. His current research focuses on robust algorithms for manifold learning and traffic prediction over complex networks.

Mr. Forero has been a recipient of the Science, Mathematics and Research for Transformation (SMART) fellowship since 2007.

Vassilis Kekatos (M'10) received the Diploma, M.Sc., and Ph.D. degrees in computer engineering and informatics from the University of Patras, Greece, in 2001, 2003, and 2007, respectively.

Since 2009, he has been a Marie Curie Fellow, and he is currently a Postdoctoral Associate with the Department of Electrical and Computer Engineering of the University of Minnesota, Minneapolis, and the Computer Engineering and Informatics Department, University of Patras, Greece. His research interests lie in the areas of statistical signal processing and learning with emphasis on the power grid, compressive sampling, and wireless communications.

Georgios B. Giannakis (F'97) received the Diploma degree in electrical engineering from the National Technical University of Athens, Greece, in 1981 and the M.Sc. degree in electrical engineering, the M.Sc. degree in mathematics, and the Ph.D. degree in electrical engineering from the University of Southern California (USC) in 1983, 1986, and 1986, respectively.

Since 1999, he has been a Professor with the University of Minnesota, where he now holds an ADC Chair in Wireless Telecommunications in the Electrical and Computer Engineering Department and serves as Director of the Digital Technology Center. His general interests span the areas of communications, networking and statistical signal processing subjects on which he has published more than 300 journal papers, 500 conference papers, 20 book chapters, two edited books, and two research monographs. Current research focuses on compressive sensing, cognitive radios, network coding, cross-layer designs, mobile ad hoc networks, wireless sensor, power, and social networks.

Dr. Giannakis is the (co-)inventor of 21 patents issued, and the (co-)recipient of seven paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G.Marconi Prize Paper Award inWireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, and the G. W. Taylor Award for Distinguished Research from the University of Minnesota. He is a Fellow of EURASIP and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.