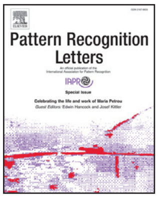
Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



Robust trimmed *k*-means

Olga Dorabiala*, J. Nathan Kutz, Aleksandr Y. Aravkin

Department of Applied Mathematics, University of Washington, Seattle, 98195, WA, USA

ARTICLE INFO

Article history:
Received 19 August 2021
Revised 20 May 2022
Accepted 12 July 2022
Available online 16 July 2022

Edited by Jiwen Lu

Keywords:
k-Means
Clustering
Robust statistics
Trimming
Unsupervised learning

ABSTRACT

Clustering is a fundamental tool in unsupervised learning, used to group objects by distinguishing between similar and dissimilar features of a given data set. One of the most common clustering algorithms is *k*-means. Unfortunately, when dealing with real-world data many traditional clustering algorithms are compromised by lack of clear separation between groups, noisy observations, and/or outlying data points. Thus, robust statistical algorithms are required for successful data analytics. Current methods that robustify *k*-means clustering are specialized for either single or multi-membership data, but do not perform competitively in both cases. We propose an extension of the *k*-means algorithm, which we call *Robust Trimmed k-means* (RTKM) that simultaneously identifies outliers and clusters points and can be applied to either single- or multi-membership data. We test RTKM on various real-world datasets and show that RTKM performs competitively with other methods on single membership data with outliers and multi-membership data without outliers. We also show that RTKM leverages its relative advantages to outperform other methods on multi-membership data containing outliers.

1. Introduction

Data science and machine learning have revolutionized the way that we do science today. Intelligent systems are used in the engineering, physical, social, and biological sciences to take in data and output, among other critical information, actionable decision making capabilities or data analyses that detail correlations between important features [1]. The three major paradigms of machine learning are supervised, unsupervised, and reinforcement learning. These three classes describe the kind of data used to structure learning tasks. In *supervised learning*, the goal is to generate a learned mapping from inputs to outputs given labeled data. The simplest example is linear regression, while a more complex example with high impact in recent years is deep neural networks. In contrast, *unsupervised learning* is used to discover the underlying patterns or structures of unlabeled data. This area includes methods for exploratory data analysis, such as dimensionality reduction and clustering. Finally, *reinforcement learning* learns how to map situations to actions, so as to maximize a numerical (delayed) reward signal [2]. The relative successes of supervised and reinforcement learning are directly related to the availability of extensive labeled data. In the absence of labels, these methods are known to perform poorly. *Semi-supervised learning* attempts to address this issue by augmenting unlabeled data with smaller portions of labeled data [1]. However, it is often infeasible or expensive to manually label even a subset of a high-dimensional dataset. In these cases, unsupervised learning techniques are the only available approach for extracting information. Such methods are compromised by lack of clear separation between features, noisy observations, and/or outlying data points and require robustification, which is what we aim to improve in the context of the common *k*-means algorithm [1].

Unsupervised learning techniques include dimensionality reduction, cluster analysis, and anomaly detection. Although often treated as separate problems, these methods have significant overlap in practice. Our specific algorithmic innovations pertain to the intersection between cluster analysis and anomaly detection. Cluster analysis seeks to divide a set of objects so as to maximize both intra-cluster similarity and inter-cluster differences, while the aim of anomaly detection is to identify outliers in the dataset. Many diverse algorithms have been developed to solve these important problems [1], including partitioning algorithms such as classic *k*-means and fuzzy *c*-means clustering [3], density based methods such as DBSCAN (density based spatial clustering of applications with noise) [4], probabilistic methods such as mixture models [1], and hierarchical clustering which produces dendrograms for data visualization. Spectral methods can extend unsupervised learning to clusters that do not have spherical and/or elliptic distributions [5].

When data is well-separated and contains no outliers, *k*-means may be able to accurately assign labels to clusters [1]. Versions

[*] Corresponding author.
E-mail address: OlgaD400@uw.edu (O. Dorabiala).

https://doi.org/10.1016/j.patrec.2022.07.007

of the the k-means algorithm date back to the mid-1950s, with seminal contributions from Steinhaus [6] and more modern versions developed by Lloyd [7] (published much later in 1982) and Forgy [8] in the mid-1960s. The k-means algorithm is simple, intuitive and can be directly applied without restrictions. Its simplicity and applicability have contributed to its appeal and widespread usage; it was named one of the top-10 algorithms in data mining in 2008 [9]. Unfortunately, real-world data may be compromised by outliers and/or complicated by simultaneous membership to multiple clusters. Under these conditions, k-means is known to perform poorly.

The poor performance of machine learning algorithms on data with corruption and noise has long been acknowledged. In the 1960s, John Tukey was the first to recognize the need for *robust* methods, coining the term *robust statistics* [10,11]. Tukey was agnostic to any particular procedure, but simply insisted that working with real data required robustification in order to stabilize the performance and predictive power of machine learning and statistical methods. Since that time, scientists have proposed numerous robustification techniques, including for clustering in unsupervised learning. These methods include data trimming [12,13], measures of outlierness [14–16], and staging methods for outlier identification [17]. We propose a novel extension of the k-means algorithm, which we call *Robust Trimmed k-means* (RTKM), that allows us to (i) capture more information than previous methods, (ii) can be used to classify both single or multi membership data, and (iii) simultaneously clusters points and identifies outliers. Experimental results show that RTKM, unlike other methods, performs competitively across all realms: on single-membership data containing outliers, multi-membership data without outliers, and multi-membership data containing outliers.

2. Related work

In this section, we review existing approaches to robustify k-means clustering. We focus on methods that build upon the basic k-means algorithm, due to k-means' simplicity, speed, and scalability [1]. These developments also apply to any algorithm that depends on k-means. An improved k-means can be integrated with methods that aim to capture the non-linearity of real world data by finding the optimal latent space representation and clustering either sequentially or simultaneously. This covers algorithms such as spectral clustering and deep clustering methods, which can be applied to a variety of data distributions on which k-means alone may not perform well.

We first review a fundamental connection between k-means clustering and optimization. The k-means algorithm can be viewed as an alternating minimization approach to solving the challenging optimization problem

$$\min_{\mathbf{c},\mathbf{W}} \sum_{j=1}^{k}\sum_{i=1}^{N} w_{j,i}||\mathbf{x}_i - \mathbf{c}_j||^2; \qquad \sum_{j=1}^{k} w_{j,i} = 1 \text{ for } i = 1:N \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \cdots \mathbf{x}_N] \in \mathbb{R}^{m\times N}$ are the data points and $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_k] \in \mathbb{R}^{m\times k}$ are the cluster centers [18]. The matrix $\mathbf{W} \in \mathbb{R}^{k\times N}$ contains auxiliary weights $w_{j,i}$ that map the point-to-cluster relationship. Each column i of $\mathbf{W}$ assigns point $\mathbf{x}_i$ to a cluster whose center is $\mathbf{c}_j$. Constraining weights to belong to the discrete set $w_{j,i} \in \{0,1\}$, (1) is a mixed integer problem equivalent to classic clustering; it is nonsmooth and nonconvex.

The simplest approach to solving the k-means problem is Lloyd's (1982) algorithm [7]. Once cluster centers are initialized, each point is alternatively assigned to its closest centroid and cluster centers are updated by taking the mean of all points in an assigned cluster, as shown in (2). In each iteration, the variables are alternatively minimized until convergence [18].

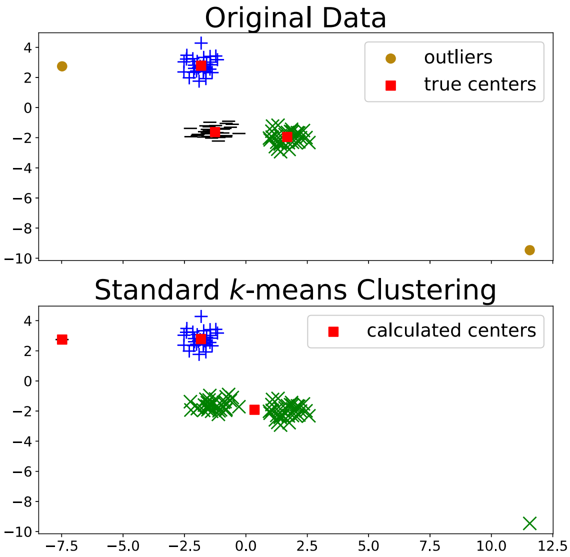


Fig. 1. (top) Original data consisting of three clusters and two outliers. (bottom) k-means incorrectly assigns a cluster center to the outlier on the right, causing two clusters to be misidentified as one. The outlier on the left skews one cluster center towards the bottom right.

$$\mathbf{c}_j^{k+1} = \frac{\sum_{i=1}^{N} w_{j,i}^k \mathbf{x}_i}{\sum_{i=1}^{N} w_{j,i}^k}; \quad w_{j,i}^{k+1} = \begin{cases} 1 & \text{if } j = \arg\min_t ||\mathbf{x}_i - \mathbf{c}_t^{k+1}||^2 \\ 0 & \text{else} \end{cases} \tag{2}$$

While k-means works well in an ideal situation, one of its main drawbacks is sensitivity to outliers and noise. Since points are classified by directly thresholding on the distance from cluster centers each iteration, outliers skew center assignment, and in turn point assignment, dramatically. Fig. 1 shows k-means inability to properly classify points in the presence of outliers.

To address this problem, numerous versions of a robust k-means algorithm have been proposed. The majority of these methods perform clustering in stages. In the first stage, the dataset is divided into clusters, and in the second stage, a measure based on the clusters is applied to the data to identify outliers. One of the earliest examples of a staged method is trimmed k-means, proposed in 1997 [19]. Trimmed k-means performs standard k-means, removes a given percentage of points with the greatest distance to their cluster centers, updates cluster centers as the mean of the remaining points in respective clusters, and repeats these steps until convergence. Ultimately, this method is ineffective, because it is unable to improve an already poor result by the standard k-means algorithm. Other staging methods that suffer from the same drawback are Outlier Removal Clustering (ORC) [17], the cluster-based local outlier factor (CBLOF) [15], the outlier factor of a cluster [14], and Local Distance-Based Outlier Factor (LDOF) [16]. In these methods, points that should be classified as outliers are masked by the standard k-means clustering.

Far fewer methods have been developed to simultaneously cluster and identify outliers. Among those that do are Outlier Detection and Clustering algorithm (ODC) [20], k-means $--$ [21], Non-exhaustive, Overlapping k-means (NEO-k-means) [22], and k-means clustering with outlier removal (KMOR) [23]. On single membership data containing outliers, KMOR outperforms all of the aforementioned methods in terms of cluster accuracy and outlier detection [23]. KMOR always produces a single-membership assignment. It identifies at most a given number of n_0 outliers by assigning them to a $k+1$th cluster based on whether they are further away than γ times the average distance between inliers and their centroids. The parameter γ is difficult to identify when the proportion of outliers is unknown, and can drastically impact the algorithm.

Of all available methods, only NEO-*k*-means, which extends *k*-means to overlapping clusters with noise, allows points to belong to multiple clusters simultaneously. On multi-membership data without outliers, NEO-*k*-means outperforms similar methods [22]. The algorithm uses a single binary weight matrix to classify points and identify outliers, where parameters σ and α give the user a way to specify the degree of overlap and proportion of outliers, respectively. Note that σ is denoted as α and α is denoted as β in the original paper. If a point is not assigned to any cluster, it is designated as an outlier. NEO-*k*-means exhaustively assigns some multiple $(1+\sigma)$ of the total number points to clusters, where $\sigma \geq 0$. Thus, it is unable to identify outliers on datasets containing only a single cluster and cannot be restricted to single-membership on data with outliers.

We propose the Robust Trimmed *k*-means (RTKM) algorithm, which can be used to classify either single- or multi-membership data in the presence of outliers and noise. In our numerical results, we therefore focus on the natural comparison between RTKM, KMOR, and NEO-*k*-means. We show that RTKM performs competitively with KMOR on single-membership data with outliers and with NEO-*k*-means on multi-membership data without outliers. Moreover, RTKM leverages its advantages in these domains to achieve superior performance on multi-membership data containing outliers.

3. Robust trimmed *k*-means

3.1. Relaxation of k-means

We propose an extension of the *k*-means objective in (3) that allows points to belong to multiple clusters and sets up a foundation for a robust extension. Multi-cluster membership is appealing, because it allows one to identify the extent of a point's membership to every cluster [24]. Instead of restricting the auxiliary weight matrix $\mathbf{W}$ to the discrete set $\{0,1\}$, we allow $\mathbf{w}_{:,i} \in \Delta_s$ so that $\sum_{j=1}^{k} w_{j,i} = s$ for all i and each $w_{j,i}$ is allowed to vary over the closed interval [0,1]. The variable s denotes the minimum number of clusters a point can belong to. When $s=1$, the objective in (3) is a classic relaxation of (1) where the weights $w_{j,i}$ can be interpreted as the probability that point i belongs to cluster j. The relaxed objective is given by

$$\min_{\mathbf{c},\mathbf{W}} \sum_{j=1}^{k}\sum_{i=1}^{N} w_{ji}||\mathbf{x}_i - \mathbf{c}_j||^2 \text{ where } \mathbf{w}_{:,i} \in \Delta_s \text{ for } i = 1:N \tag{3}$$

which is a continuous optimization problem, still nonsmooth and nonconvex. Problem (3) can also be solved using alternating minimization. The centers are updated as in standard *k*-means using the Gauss-Seidel step in (2). To better control how quickly the weights are updated, we use the Proximal Alternating Minimization (PAM) approach, which can be thought of as a proximal regularization of the Guass-Seidel scheme [25]. The update for $\mathbf{W}$ is given in (4). PAM is guaranteed to converge as long as the step size $d_k > 1$ and gives the practitioner additional control in managing the weight update in a rigorous way.

$$\mathbf{w}^{k+1}_{[:,i]} = \text{proj}_{\Delta_s}\left(\mathbf{w}^k_{:,i} - \frac{1}{d_k}||\mathbf{x}_i - \mathbf{C}^{k+1}||^2\right) \tag{4}$$

Solving the relaxed *k*-means objective allows for greater modeling flexibility. In *k*-means, the columns of the weight matrix are projected onto the vertices of the capped simplex each iteration, whereas in the relaxation, the weights can take on a continuum of values in the capped simplex. They can optionally be projected onto the vertices during cluster assignment after the algorithm has already converged. Fig. 2 gives a glimpse into how the clustering process works. Point colors existing on the gradient between blue and green represent the continuous cluster weights. In iteration 0, the weights are randomly assigned and cluster centers are initialized at random. By iteration 3, the centers begin to drift apart, and two clusters begin to form. Points on the boundary remain on the gradient between blue and green. In iteration 5, cluster centers begin to stabilize, and in iteration 20, relaxed *k*-means converges, and all the weights are on the vertices of the 1-capped simplex. The relaxation of weighted *k*-means allows us to quantify the extent of membership of a point to each cluster at every iteration.

3.2. Robust trimmed k-means

In order to create a robust *k*-means method, we model outlier detection using an analogous approach taken in Section 3.1 for multi-cluster membership. In this way, we avoid having to define our own measure of "outlierness". Our proposed method, which we call Robust Trimmed *k*-means (RTKM), simultaneously classifies points and identifies outliers by minimizing the objective function given in Eq. (5).

$$\min_{\mathbf{c},\mathbf{v},\mathbf{W}} \sum_{i=1}^{N} v_i \sum_{j=1}^{k} w_{j,i}||\mathbf{x}_i - \mathbf{c}_j||^2 \,\text{where}\, \mathbf{w}_{:,i} \in \Delta_s \,\text{and}\, \mathbf{v} \in \Delta_{N-[\alpha N]} \tag{5}$$

As before, $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{m\times N}$ are the data points, $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_k] \in \mathbb{R}^{m\times k}$ are the cluster centers, and $\mathbf{W} \in \mathbb{R}^{k\times N}$ is a matrix of weights with each column in the *s*-capped simplex. The variable $\mathbf{v} \in \mathbb{R}^N$ is a vector that identifies outliers and belongs to the $N-[\alpha N]$-capped simplex, where α is a given proportion of expected outliers. Here $[\cdot]$ denotes the nearest integer, where we round up for half integer values. The constraint on $\mathbf{v}$ ensures that $N-[\alpha N]$ points are designated as inliers, since $\sum_{i=1}^{N} v_i = N - [\alpha N]$. Similarly, the constraint on the columns of $\mathbf{W}$ ensures that every point is assigned to at least s clusters, since $\sum_{j=1}^{k} w_{j,i} = s$. As discussed in Section 3.1, this constraint allows for each point $\mathbf{x}_i$ to belong to more than one cluster. To enforce single cluster membership, we set $s=1$ and make final assignments by calculating the arg max over each column of $\mathbf{W}$ once the algorithm converges. The objective in Eq. (5) is approximately solved using Algorithm 1. We alternately minimize the objective with respect to three variables: $\mathbf{W}$, $\mathbf{c}$, and $\mathbf{v}$. We use a Gauss-Seidel update for the centers and Proximal Alternating Minimization (PAM) [25] updates for both the columns of $\mathbf{W}$ and for $\mathbf{v}$, similar to the process described in Section 3.1. Algorithm 1 is guaranteed to converge as long as $e_k > 1$ and $d_k > 1$. In practice, we set $e_k = d_k = 1.1$.

Algorithm 1 Robust Trimmed *k*-means (RTKM).

```
procedure RTKM($\mathbf{X}, k, \alpha, s$)                ▷ Input $\mathbf{X}$, $k$, $\alpha$, and $s$
    Initialize $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_k]$, $e_k = 1.1$, and $d_k = 1.1$
    while not converged do
        $\mathbf{c}^{k+1}_{:,j} = \sum_{i=1}^{N} v^k_i w^k_{j,i}\mathbf{x}_i / \sum_{i=1}^{N} v^k_i w^k_{j,i}$
        $\mathbf{w}^{k+1}_{:,i} = \text{proj}_{\Delta_s}\left(\mathbf{w}^k_{:,i} - \frac{1}{d_k} v^k_i ||\mathbf{x}_i - \mathbf{C}^{k+1}||^2\right)$
        $v^{k+1} = \text{proj}_{\Delta_{[\alpha N]}}\left(v^k - \frac{1}{e_k}\sum_{j=1}^{k} \mathbf{w}_{j,:k=1}||\mathbf{X} - \mathbf{c}^{k+1}_j||^2\right)$
    if s = 1 then
        $\mathbf{w}_{:,i} = \arg\max_j \mathbf{w}_{j,i}$
    else
        $\mathbf{w}_{:,i} = \max(\mathbf{w}_{:,i}, 0)$
    return $\mathbf{C}, \mathbf{w}, \mathbf{v}$
```

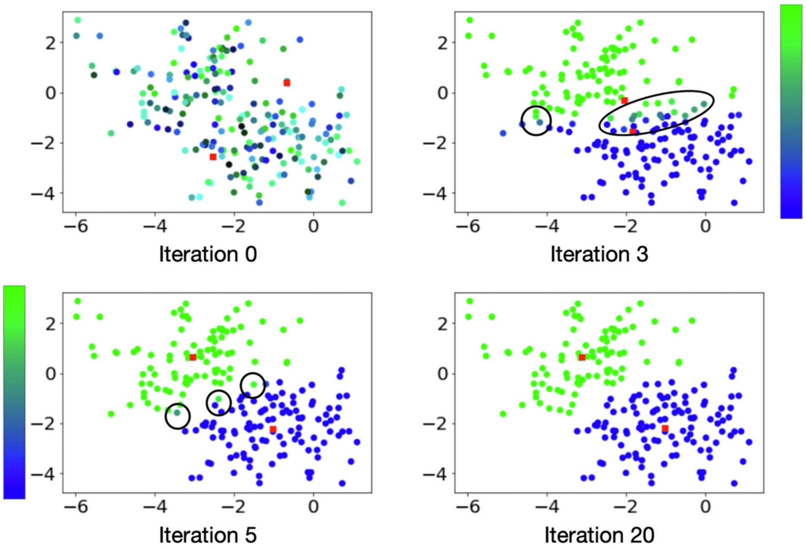


Fig. 2. Clustering process of relaxed k-means. The colors of the points represent the continuum of weights that assign points to clusters. In iteration 0, all weights and cluster centers are assigned randomly. In iteration 3, two distinct clusters begin to form. In iteration 5, cluster centers begin to stabilize and points on the boundary retain partial membership to both clusters. In iteration 20, relaxed k-means converges to two distinct clusters.

The algorithm takes as input the data $\mathbf{X}$, the number of clusters k, and parameters α and s, which control the number of outliers and number of members in each cluster, respectively. Centroids can be initialized using any scheme. If we know the percentage of outliers in the data, we set α equal to that value. Likewise, if we know the cardinality of the data, we set $s = [\text{cardinality}]$, where $[\cdot]$ denotes the closest integer and we round up for half-integer values. Currently, there is no principled approach to estimating these parameters. However, we demonstrate that RTKM performs competitively with other methods on data containing outliers using a range of α values.

4. Experiments

We compare the performance of RTKM against other methods that simultaneously perform clustering and outlier detection, specifically KMOR [23] and NEO-k-means [22]. We focus our comparison against KMOR and NEO-k-means, because these methods were shown to outperform others on various datasets. KMOR [23] outperformed ODC [20], k-means−− [21] and NEO-k-means [22] on single-membership data containing outliers, while NEO-k-means [22] outperformed six other methods on multi-membership data without outliers. We evaluate the performance of RTKM on three types of datasets: single-membership datasets containing outliers, multi-membership datasets without outliers, and mutli-membership datasets with outliers.

The quality of the cluster assignments is measured using the metric in [22], average F_1 score, which quantifies how well each algorithm finds the ground truth clusters. The F_1 score is defined as $F_1 = TP/(TP + 0.5(FP + FN))$ where TP denotes true positives, FP denotes false positives, and FN denotes false negatives. This metric ranges from 0 to 1, with values closer to 1 implying better classification. To calculate the average F_1 score, predicted clusters are matched to ground-truth clusters so that the average F_1 score among all clusters is maximized. Outliers are considered their own cluster for the purpose of calculating the average F_1 score.

The ability to correctly identify outliers is measured using $M_e = \sqrt{(FP_{\text{rate}})^2 + (1 - TP_{\text{rate}})^2}$ from Gan and Ng [23], where TP_{rate} and FP_{rate} denote ratios of true positives to all positives and true negatives to all negatives. The M_e score depends only on the true and identified outliers in the dataset. The value of M_e ranges from 0 to $\sqrt{2}$, with better outlier classifiers near 0.

For data containing outliers, we test the performance of each method in terms of both F_1 and M_e score with various values for the expected percentage of outliers to see how sensitive results are to parameter choice. For every value of α, we complete 50 runs of an algorithm with different cluster center initialization each time. Performance metrics are reported as the minimum, maximum, and average M_e and F_1 scores over the 50 runs.

4.1. Single-membership data with outliers

We begin by evaluating RTKM against KMOR and NEO-k-means on two single-membership datasets containing outliers. The first is the Breast Cancer Wisconsin (WBC) dataset [26] from UCI Machine Learning Repository [27]. The WBC dataset contains 699 instances of tumors with 9 attributes each, all of which are classified as benign or malignant, with the latter being treated as outliers.

Fig. 3 shows performance metrics for the three algorithms over a range of α values on the WBC dataset. RTKM and KMOR exhibit almost identical performance when given the same parameters, while NEO-k-means, by design, is unable to identify any outliers when $k = 1$.

Next, we test the three algorithms on the shuttle training dataset [27]. The data contains 43,500 records and 7 classes, described by 9 numerical features. The three largest classes contain 99.57% of the data, so we consider these classes to be inliers and the remaining four to be outliers, as in [23]. We again test the sensitivity of the results for RTKM, KMOR, and NEO-k-means to choice of α. Fig. 4 shows that while $\alpha \leq 0.02$, all three methods perform similarly. Beyond this point, increasing α leads RTKM and NEO-k-means to identify more outliers correctly and achieve much lower average M_e scores compared to KMOR. Unfortunately, the corresponding increase in false positive outliers leads to comparatively lower average F_1 scores. In applications where there is a high cost associated with false positive outliers, KMOR may be preferential. Conversely, in applications were the goal is to identify as many outliers as possible, RTKM is superior.

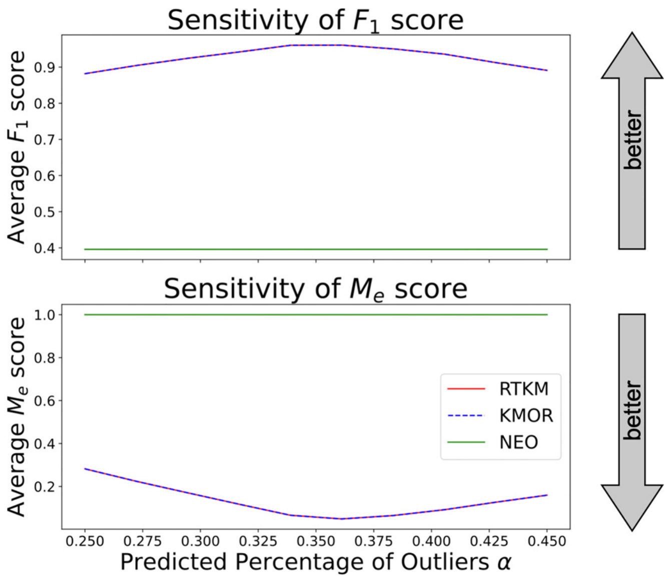


Fig. 3. Sensitivity of RTKM, KMOR, and NEO-k-means to choice of α on the WBC dataset. RTKM and KMOR exhibit almost identical performance. NEO-k-means is unable to identify any outliers. (Parameters: $k = 1$, RTKM $s = 1$, KMOR $\gamma = 1$, NEO-k-means $\sigma = 0$).

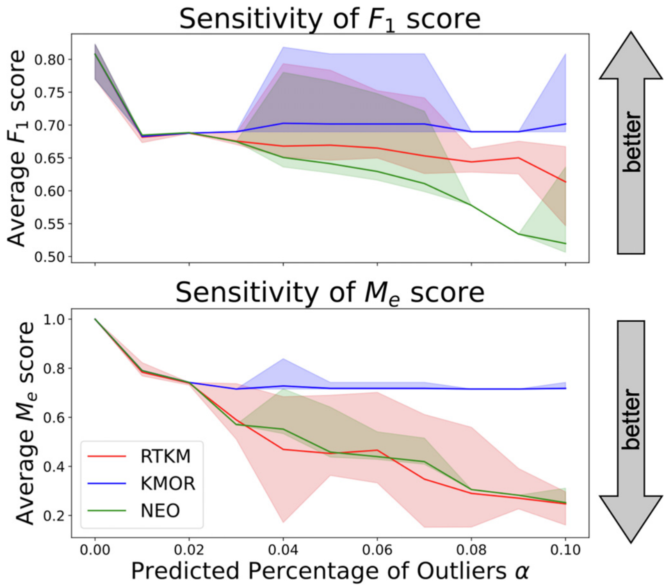


Fig. 4. Sensitivity of RTKM, KMOR, and NEO-k-means to choice of α on the Shuttle dataset. All three methods perform similarly until $\alpha \geq 0.02$. After, KMOR and NEO-k-means identify significantly more outliers correctly, but the increase in false positives leads to lower average F_1 scores. (Parameters: $k = 14$, RTKM $s = 1$, KMOR $\gamma = 1$, NEO-k-means $\sigma = 0$).

4.2. Multi-membership data without outliers

Unlike KMOR, RTKM and NEO-*k*-means are able to classify points as belonging to multiple clusters [22]. The parameters in NEO-*k*-means determine how many point assignments are made and at most how many outliers are identified. In contrast, RTKM's parameters determine at least how many assignments are made and at least how many outliers are identified.

We evaluate the three methods on the "yeast", "scene", and "emotions" datasets from Tsoumakas et al. [28]. The "yeast" dataset contains 2417 instances with 103 numerical attributes. There are 14 classes, and the cardinality is 4.237. The "scene" dataset contains 2407 instances with 294 numeric attributes. There are 6 classes, and the cardinality is 1.074. Finally, "emotions" contains 593 instances with 72 numerical attributes. There are 6 classes, and the cardinality is 1.869. Both "yeast" and "scene" are used for testing MOC [29], fuzzy *k*-means [24], explicit sparsity constrained clustering (esp) [30], implicit sparsity constrained clustering (isp) [30], OKM [31], and restricted OKM (rokm) [32] in [22]. We borrow performance reports on "yeast" and "scene" for all aforementioned algorithms from [22]. Results on "emotions" are only reported for RTKM, KMOR, and NEO-*k*-means. Each algorithm is run five times using the same cluster center initialization, and the result that leads to the best objective function value is chosen, as in [22].

Average F_1 scores for each algorithm are shown in Table 1. NEO-*k*-means achieves the highest F_1 score on "scene" and "yeast", but in both cases the F_1 score of RTKM is the second highest of all scores reported. On "emotions", RTKM achieves the highest F_1 score of the three methods. We note that on "scene", KMOR performs just as well as RTKM due to the cardinality of "scene" being so close to one. Contrastingly, on "yeast" and "emotions", KMOR is unable to produce a competitive result due to the higher cardinalities and the fact that the method is not designed to run on multi-membership data.

4.3. Multi-membership data with outliers

In order to compare RTKM against KMOR and NEO-*k*-means on multi-membership data containing outliers, we add noise in two different ways to "scene", "yeast", and "emotions" from Section 4.2. The first approach is to add Gaussian noise, while the second is to take the average of the data points and add to it some multiple of the average of their standard deviations.

We begin by adding 100 noise points to "scene", so that it contains ∼ 4% outliers. Fig. 5a and b demonstrate the performance of the three methods over various α values for the Gaussian and alternative noise models, respectively. On both datasets, RTKM and KMOR score similarly and outperform NEO-*k*-means in terms of average F_1 scores. KMOR's competitive performance here is unsurprising, given that the original "scene" data has a cardinality close to 1. Fig. 5 a shows that all three methods perform almost identically in terms of M_e scores on "scene" + Gaussian noise, while Fig. 5 b shows that RTKM and KMOR are able to achieve slightly better outlier detection than NEO-*k*-means on "scene" + noise.

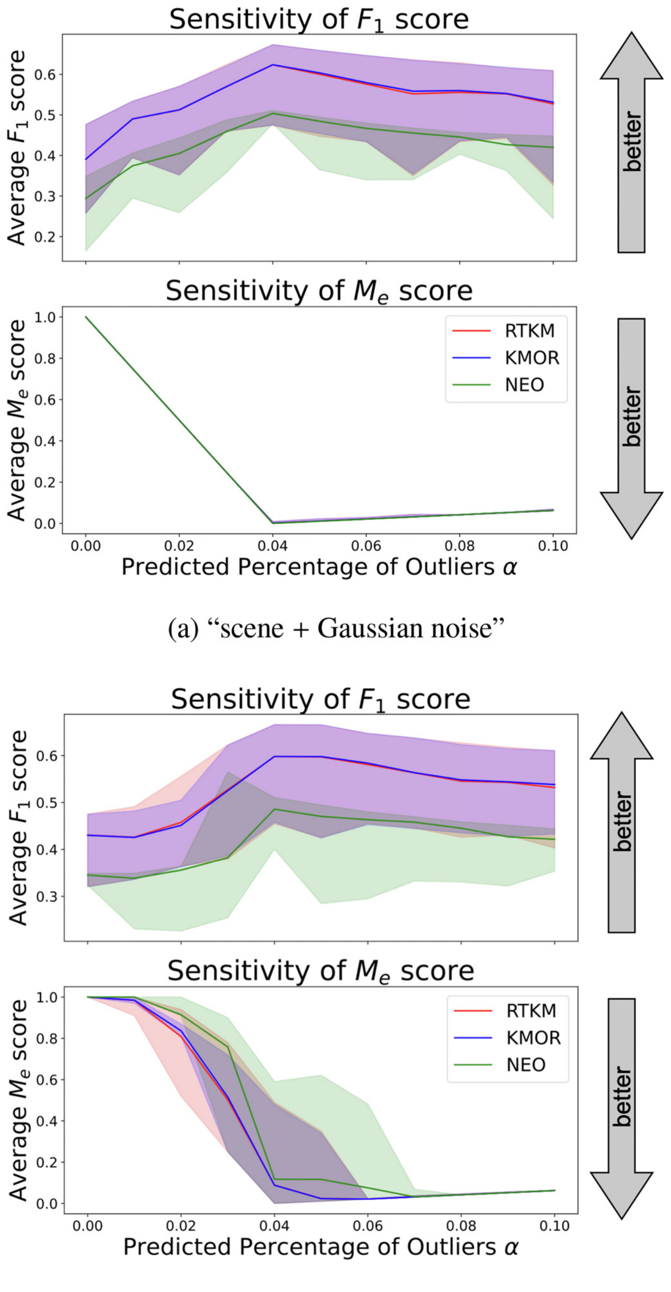
Next, we add 150 noise points to "yeast" so that the data contains ∼ %6 outliers. We again test the sensitivity of the results of all three methods to α for "yeast" with added Gaussian and alternative noise. As seen in Fig. 6a and b, of the three methods tested, RTKM achieves the highest average F_1 scores over all values of α, with NEO-*k*-means following closely behind. KMOR scores poorly for classification accuracy, because of the higher cardinality of the data. In terms of outlier detection, all three methods score similarly when Gaussian noise is added, but on average, RTKM substantially outperforms the other methods with the addition of alternative noise.

Finally, we add 25 noise points to "emotions" so that the data contains ∼ 4% outliers. Fig. 7a and b show the sensitivity of each of the methods to α for the addition of Gaussian and alternative noise. On both datasets, RTKM achieves the highest average F_1 scores over all values of α, followed closely by NEO-*k*-means. Again with the addition of Gaussian noise, all three methods perform similarly in terms of outlier detection, with RTKM scoring slightly better. On "emotions" + noise, RTKM achieves by far the lowest average M_e scores.

On multi-membership data with outliers, RTKM achieves the highest average F_1 scores across all datasets tested. On the lower

Table 1
Average F_1 scores of various multi-membership clustering methods on the “yeast”, “scene”, and “emotions” datasets. NEO-k-means achieves the best F_1 score on “yeast” and “scene”. However, RTKM achieves the second highest F_1 score on both datasets, demonstrating its competitiveness. On “emotions”, RTKM achieves the highest F_1 score, followed closely by NEO-k-means. (“yeast” parameters: $k = 14, \alpha = 0$, RTKM $s = 4$, KMOR $\gamma = 9$; “scene” parameters: $k = 6, \alpha = 0$, RTKM $s = 1$, KMOR $\gamma = 9$; “emotions” parameters: $k = 6, \alpha = 0$, RTKM $s = 2$, KMOR $\gamma = 9$, NEO-k-means $\sigma = 1$) .

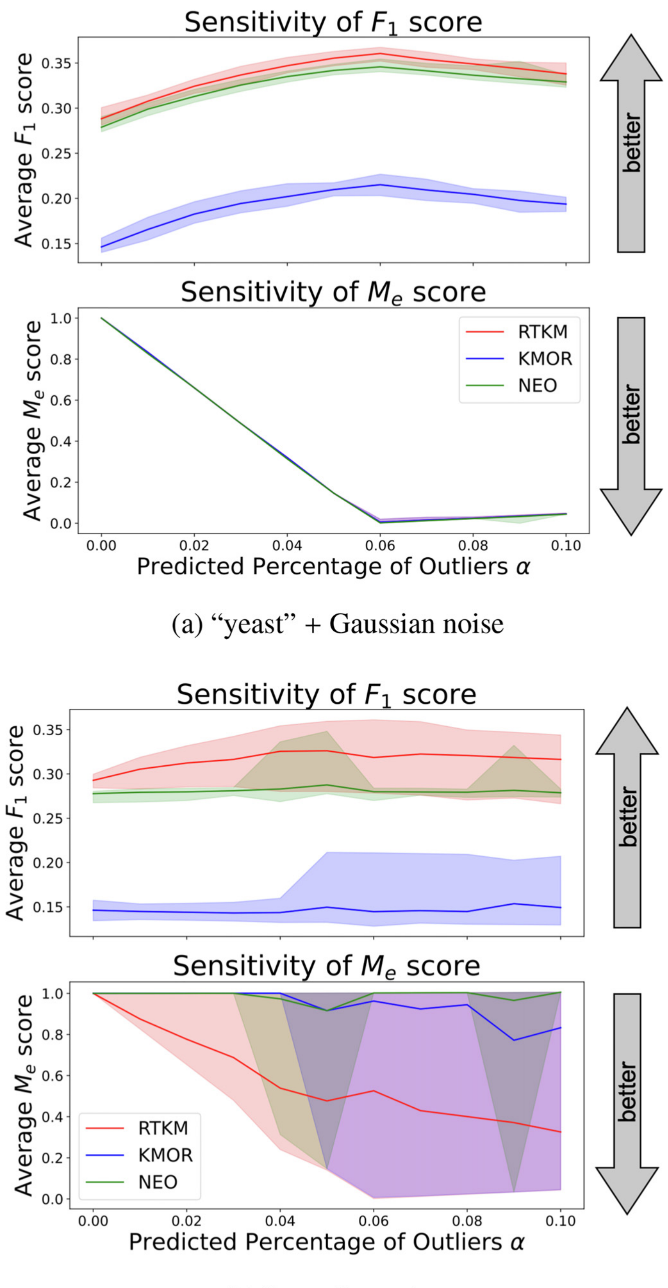
	RTKM	KMOR	NEO-k-means	moc	fuzzy	esp	isp	okm	rokm
“yeast”	0.325	0.161	0.366	-	0.308	0.289	0.203	0.311	0.203
“scene”	0.598	0.598	0.626	0.467	0.431	0.572	0.586	0.571	0.593
“emotions”	0.399	0.311	0.373	-	-	-	-	-	-



(a) “scene + Gaussian noise”

(b) “scene” + noise

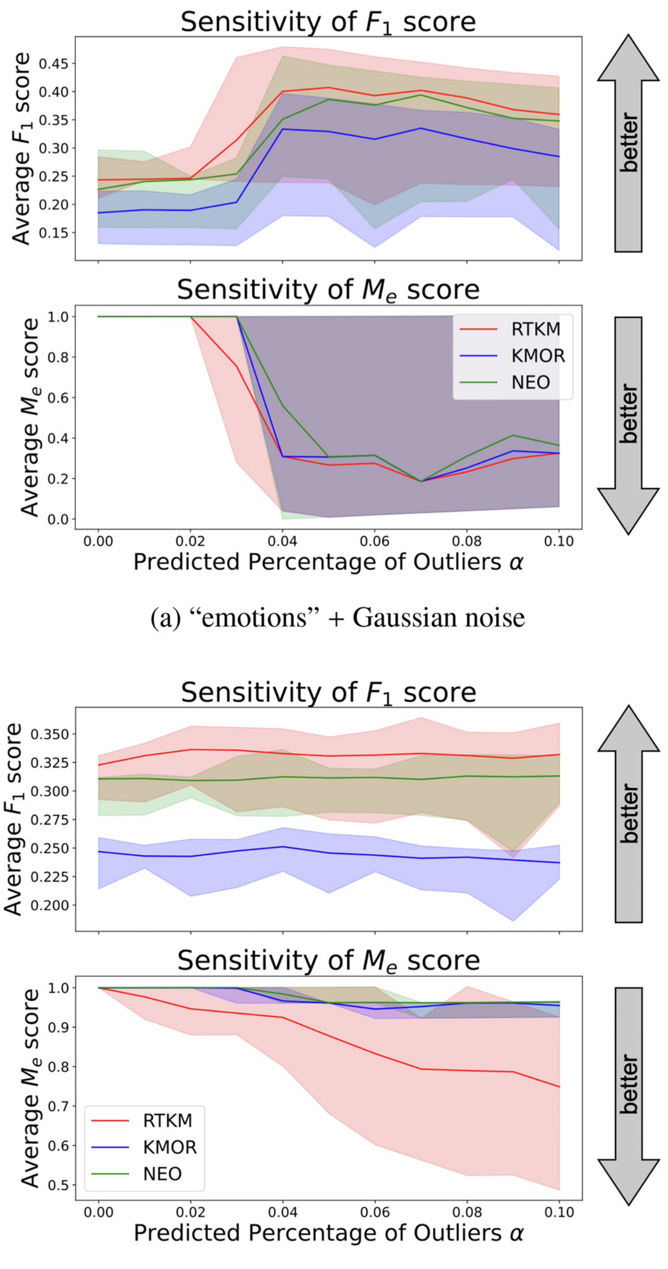
Fig. 5. Sensitivity of RTKM, KMOR, and NEO-k-means to choice of α on the (a) “scene” + Gaussian noise and (b) “scene” + noise datasets. In both (a) and (b), RTKM and KMOR perform similarly well and outperform NEO-k-means in terms of average F_1 scores. In (a), all three methods perform almost identically in terms of outlier detection, while in (b), RTKM and KMOR attain lower average M_e scores than NEO-k-means. (Parameters: $k = 6$, RTKM $s = 2$, KMOR $\gamma = 1$, NEO-k-means $\sigma = 1$).



(a) “yeast” + Gaussian noise

(b) “yeast” + noise

Fig. 6. Sensitivity of RTKM, KMOR, and NEO-k-means to choice of α on the (a) “yeast” + Gaussian noise and (b) “yeast” + noise datasets. In both (a) and (b), RTKM performs best in terms of F_1 score, followed by NEO-k-means and KMOR. In (a), all three methods perform almost identically in terms of outlier detection, while in (b), RTKM obtains a noticeably lower M_e score. (Parameters: $k = 14$, RTKM $s = 4$, KMOR $\gamma = 1$, NEO-k-means $\sigma = 3$).



(a) “emotions” + Gaussian noise

(b) “emotions” + noise

Fig. 7. Sensitivity of RTKM, KMOR, and NEO-k-means to choice of α on the (a) “emotions” + Gaussian noise and (b) “emotions” + noise datasets. In both (a) and (b), RTKM performs best in terms of average F_1 scores. In (a), all three methods perform similarly, though RTKM maintains a lower average M_e scores over all values of α, while in (b) RTKM obtains noticeably lower M_e scores on average than either other method. (Parameters: $k = 6$, RTKM $s = 2$, KMOR $\gamma = 1$, NEO-k-means $\sigma = 1$).

cardinality “scene”-based data, KMOR classifies points just as well, while on the higher cardinality “yeast”- and “emotions”-based data, KMOR is not competitive and NEO-*k*-means performs second-best. Unsurprisingly, all three methods perform outlier detection well when the added noise is normally distributed. On “scene” + and “yeast” + Gaussian noise, the average M_e scores are almost indistinguishable, while on “emotions” + Gaussian noise, RTKM achieves only marginally lower M_e scores. It is when alternative noise is added to the data that RTKM’s superior ability to detect outliers is pronounced.

Table 2
Ranks of RTKM, KMOR, and NEO-k-means in terms of F_1 score.

	RTKM	KMOR	NEO-k-means
WBC	1.5	1.5	3.0
Shuttle	2.0	1.0	3.0
“yeast”	2.5	2.5	1.0
“scene”	2.0	3.0	1.0
“emotions”	1.0	3.0	2.0
“yeast” + Gaussian noise	1.0	3.0	2.0
“scene” + Gaussian noise	1.5	1.5	3.0
“emotions” + Gaussian noise	1.0	3.0	2.0
“yeast” + noise	1.0	3.0	2.0
“scene” + noise	1.5	1.5	3.0
“emotions” + noise	1.0	3.0	2.0
Average Rank	1.45	2.36	2.18

Table 3
Ranks of RTKM, KMOR, and NEO-k-means in terms of M_e score.

	RTKM	KMOR	NEO-k-means
WBC	1.5	1.5	3.0
Shuttle	1.0	3.0	2.0
“yeast” + Gaussian noise	2.0	2.0	2.0
“scene” + Gaussian noise	2.0	2.0	2.0
“emotions” + Gaussian noise	1.0	2.0	3.0
“yeast” + noise	1.0	2.0	3.0
“scene” + noise	1.5	1.5	3.0
“emotions” + noise	1.0	2.0	3.0
Average Rank	1.38	2.00	2.63

5. Nonparametric statistical validation of clustering

We use the Friedman test method [33] to perform a nonparametric statistical validation of the results of RTKM, KMOR, and NEO-*k*-means. We calculate a representative F_1 and M_e score for each method on every dataset by averaging the values of F_1 and M_e for all runs over all values of α. Table 2 lists the ranks of these three methods on all datasets in terms of F_1 score, while Table 3 lists the ranks in terms of M_e score.

For F_1 score, the Q-value is below the critical value of the Friedman test with 11 datasets, meaning that the difference of RTKMs performance in terms of classification accuracy is not significant. On the other hand, the Q-value for M_e score is 6.25, which is the critical value of the Friedman test for an α level of 0.05 when the number of datasets is 8. Carrying out a post-hoc Nemenyi test [34] to quantitatively evaluate the differences between methods, we find that RTKM significantly outperforms NEO-*k*-means, but not KMOR in terms of outlier identification. Although the differences in the F_1 scores of the three methods and the M_e scores of RTKM and KMOR are not deemed significant, the overall performance of RTKM in terms of both F_1 and M_e scores is the best, as evidenced by the lowest average rank of RTKM in both Tables 2 and 3.

6. Conclusions and future work

We propose Robust Trimmed *k*-means (RTKM) as an algorithm for simultaneous point classification and outlier detection that can operate on both single- and multi-membership data. The parameters *k* and α control the number of clusters and expected percentage of outliers respectively. The parameter *s* controls at least how many clusters each point belongs to. Methods such as X-means [35] and G-means [36] have addressed the problem of estimating the number of clusters by running *k*-means repeatedly with different values of *k* until some criteria is satisfied. Future work could address how these methods could be altered to use our objective function to estimate *k*. At the moment, there is no principled approach to estimating the other two parameters α and *s*. We leave the investigation of such an approach for future work and demonstrate that RTKM remains competitive with existing methods over a range of α values.

The innovations presented rely on a robust relaxed formulation for the weighted k-means algorithm that allows the classification weight matrix to exist on a continuum of values [0,1], rather than the binary set $\{0, 1\}$. This relaxation gives the user a way to track the extent of membership of a point to each cluster at every iteration and provides flexibility for multi-cluster membership. We apply the same methodology for outlier detection, thereby avoiding explicitly defining a measure of "outlierness". Relaxation-based formulations have proven to be effective in a number of recent applications [37,38]. In the context of the current application, relaxation to a continuum of values, coupled with the PAM algorithm, provides a way to search the model space more effectively to discover clusters and outliers.

We test RTKM on three types of data: single-label data with outliers, multi-label data without outliers, and mutli-label data with outliers. While KMOR and NEO-k-means set the benchmark for the first two types of data, respectively, they do not perform well on both. RTKM remains competitive on both types of data, and outperforms both KMOR and NEO-k-means on multi-label data with outliers. On single-label data, we demonstrate that NEO-k-means is, by design, unable to perform when $k = 1$ and when $k > 1$, RTKM and NEO-k-means achieve improved outlier detection over KMOR at the cost of lower average F_1 scores. We conclude that in applications where the cost of false positive outliers is high, KMOR may be preferred, but in applications where it is important to identify as many outliers as possible, RTKM may be favored. On multi-label data without outliers, RTKM remains competitive against NEO-k-means, achieving a higher F_1 score on two datasets than every other method NEO-k-means is compared against in [22] and even scoring higher in classification accuracy than NEO-k-means on a third dataset. Since KMOR does not have the functionality to make multi-membership assignments, it is unable to remain competitive when the cardinality of a dataset is greater than or equal to 1.5. On multi-label data containing outliers, we show that RTKM consistently scores highest in terms of average classification accuracy over all values of α. Further, when the noise added to the data is normally distributed, all three methods perform outlier detection almost identically well, but when the added noise follows an alternative model, RTKM clearly achieves the lowest average M_e scores. Finally, we use the Friedman test method to statistically validate the results of the clustering methods. We find that RTKM's outlier detection significantly outperforms that of NEO-k-means. Although the differences in the three methods' classification accuracies and the outlier detection of RTKM and KMOR are not deemed significant, the overall performance of RTKM is still superior.

Code Availability: https://github.com/OlgaD400/Robust-Trimmed-K-Means

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[2] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[3] S. Askari, Fuzzy c-means clustering algorithm for data with unequal cluster sizes and contaminated with noise and outliers: review and development, Expert Syst. Appl. 165 (2021) 113856.
[4] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, vol. 96, 1996, pp. 226–231.
[5] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2002, pp. 849–856.
[6] H. Steinhaus, Sur la division des corps matériels en parties, Bull. Acad. Polon. Sci 1 (804) (1956) 801.
[7] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137.
[8] E.W. Forgy, Cluster analysis of multivariate data: efficiency versus interpretability of classifications, Biometrics 21 (1965) 768–769.
[9] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, S.Y. Philip, et al., Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2008) 1–37.
[10] P.J. Huber, John W. Tukey's contributions to robust statistics, Ann. Stat. (2002) 1640–1648.
[11] D. Donoho, 50 Years of data science, J. Comput. Graph. Stat. 26 (4) (2017) 745–766.
[12] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, vol. 589, John wiley & sons, 2005.
[13] A. Aravkin, D. Davis, Trimmed statistical estimation via variance reduction, Math. Oper. Res. 45 (1) (2020) 292–322.
[14] S.-y. Jiang, Q.-b. An, Clustering-based outlier detection method, in: 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 2, IEEE, 2008, pp. 429–433.
[15] Z. He, X. Xu, S. Deng, Discovering cluster-based local outliers, Pattern Recognit. Lett. 24 (9–10) (2003) 1641–1650.
[16] K. Zhang, M. Hutter, H. Jin, A new local distance-based outlier detection approach for scattered real-world data, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2009, pp. 813–822.
[17] V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, P. Fränti, Improving k-means by outlier removal, in: Scandinavian Conference on Image Analysis, Springer, 2005, pp. 978–987.
[18] J.Z. Huang, M.K. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[19] J.A. Cuesta-Albertos, A. Gordaliza, C. Matrán, et al., Trimmed k-means: an attempt to robustify quantizers, Ann. Stat. 25 (2) (1997) 553–576.
[20] M. Ahmed, A.N. Mahmood, A novel approach for outlier detection and clustering improvement, in: 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2013, pp. 577–582.
[21] S. Chawla, A. Gionis, k-means–: A unified approach to clustering and outlier detection, in: Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM, 2013, pp. 189–197.
[22] J.J. Whang, I.S. Dhillon, D.F. Gleich, Non-exhaustive, overlapping k-means, in: Proceedings of the 2015 SIAM International Conference on Data Mining, SIAM, 2015, pp. 936–944.
[23] G. Gan, M.K.-P. Ng, K-means clustering with outlier removal, Pattern Recognit. Lett. 90 (2017) 8–14.
[24] J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, IEEE Trans. Pattern Anal. Mach. Intell. (1) (1980) 1–8.
[25] H. Attouch, J. Bolte, P. Redont, A. Soubeyran, Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Łojasiewicz inequality, Math. Oper. Res. 35 (2) (2010) 438–457.
[26] O.L. Mangasarian, W.H. Wolberg, Cancer Diagnosis via Linear Programming, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 1990.
[27] D. Dua, C. Graff, UCI machine learning repository, 2017. http://archive.ics.uci.edu/ml.
[28] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, I. Vlahavas, Mulan: a java library for multi-label learning, J. Mach. Learn. Res. 12 (2011) 2411–2414.
[29] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, R.J. Mooney, Model-based overlapping clustering, in: Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 2005, pp. 532–537.
[30] H. Lu, Y. Hong, W.N. Street, F. Wang, H. Tong, Overlapping clustering with sparseness constraints, in: 2012 IEEE 12th International Conference on Data Mining Workshops, IEEE, 2012, pp. 486–494.
[31] G. Cleuziou, An extended version of the k-means method for overlapping clustering, in: 2008 19th International Conference on Pattern Recognition, IEEE, 2008, pp. 1–4.
[32] C.-E. ben N'Cir, G. Cleuziou, N. Essoussi, Identification of non-disjoint clusters with small and parameterizable overlaps, in: 2013 International Conference on Computer Applications Technology (ICCAT), IEEE, 2013, pp. 1–6.
[33] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J Am Stat Assoc 32 (200) (1937) 675–701.
[34] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[35] D. Pelleg, A.W. Moore, et al., X-means: extending k-means with efficient estimation of the number of clusters, in: ICML, vol. 1, 2000, pp. 727–734.
[36] Z. Zhao, S. Guo, Q. Xu, T. Ban, G-means: a clustering algorithm for intrusion detection, in: International Conference on Neural Information Processing, Springer, 2008, pp. 563–570.
[37] P. Zheng, T. Askham, S.L. Brunton, J.N. Kutz, A.Y. Aravkin, A unified framework for sparse relaxed regularized regression: SR3, IEEE Access 7 (2018) 1404–1423.
[38] K. Champion, P. Zheng, A.Y. Aravkin, S.L. Brunton, J.N. Kutz, A unified sparse optimization framework to learn parsimonious physics-informed models from data, IEEE Access 8 (2020) 169259–169271.