Knowledge-Based Systems 210 (2020) 106518

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Robust multi-view k-means clustering with outlier removal

Chuan Chen [a,b,*], Yu Wang [a], Weibo Hu [a], Zibin Zheng [a,b]

[a] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[b] National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China

ARTICLE INFO

Article history:
Received 17 March 2020
Received in revised form 8 October 2020
Accepted 11 October 2020
Available online 15 October 2020

Keywords:
Multi-view clustering
Robust clustering
K-means
Outlier detection

ABSTRACT

Contemporary datasets are often comprised of multiple views of data, which provide complete and complementary information in different views, and multi-view clustering is one of the most crucial techniques in multi-view data analysis. However, traditional multi-view clustering methods are sensitive to noises and outliers, suffering from severe performance degradation when the dataset contains many outliers. Moreover, the commonly used multi-view clustering methods are restricted by high time complexity. To address these problems, we propose a robust multi-view k-means algorithm with outlier detection, i.e., Multi-View Clustering with Outlier Removal (MVCOR). This method is designed to remove the outliers and thus boosts the clustering performance on multi-view data with low time complexity. By defining two types of outliers, MVCOR uses the well-defined outlier removal strategy to categorize all the outliers into two specific clusters and performs robust clustering on the clean data at the same time. This strategy significantly improves the clustering performance as well as the model robustness, making MVCOR a more practical approach for real-world scenarios. Besides, the proposed model is efficiently optimized by a well-designed alternating minimization algorithm which is strictly proved to be convergent. Extensive experiments on both synthetic and real-world datasets demonstrate that MVCOR consistently outperforms the related clustering methods on clustering performance as well as robustness to outliers, and achieves comparable performance to the state-of-the-art multi-view outlier detection methods.

1. Introduction

Multi-view data is usually represented by different heterogeneous but related groups of features, where each group of features, or each view, is collected from diverse sources or obtained from various domains. For example, as shown in Fig. 1(a), multiple cameras focus on the same object from different angles, and the photos provided by different cameras are seen as different views. Another example is Fig. 1(b), on the web page, an item is described by the corresponding web image and its description, which provide visual and textual information respectively. For these scenarios, any particular single-view features cannot comprehensively describe the instance [1], so it is essential to take full advantage of the abundant information in multi-view data. However, if we just crudely concatenate all views of data and then use the single-view algorithm to process them, the statistical characteristics in each view and the relations among all views will be ignored [2], and the clustering performance may not even be improved since the multiple views may come from different representation spaces [3,4] or some of the views contain noisy or irrelevant information [5]. To fully utilize the multi-view data, we regard each view as particular information and process them by an appropriate multi-view algorithm which integrates different data representations without losing data information [6].

In recent years, several promising multi-view clustering methods have been proposed. Among them, the graph-based method is representative and often performs well by utilizing data graph and manifold information [7,8]. However, the graph-based method suffers from high time complexity due to the affinity graph construction and eigen-decomposition [9], and it is also often difficult to interpret [10]. On the one hand, most existing methods, e.g., subspace clustering, which performs graph learning and spectral clustering separately [11], face the same problem. On the other hand, the performance of many traditional multi-view clustering methods is easily affected by outliers or noise [12,13]. Such drawbacks make it difficult to apply these methods to some real-world scenarios, and there are many variants try to fix it. To avoid the high time complexity, many methods consider adopting the k-means algorithm to perform clustering due to its simplicity and effectiveness [14,15], so these methods can be used to processing the large-scale data [16]. Moreover, the k-means based

* Corresponding author at: School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China.
E-mail addresses: chenchuan@mail.sysu.edu.cn (C. Chen), wangy475@mail2.sysu.edu.cn (Y. Wang), huwb7@mail2.sysu.edu.cn (W. Hu), zhzibin@mail.sysu.edu.cn (Z. Zheng).

https://doi.org/10.1016/j.knosys.2020.106518

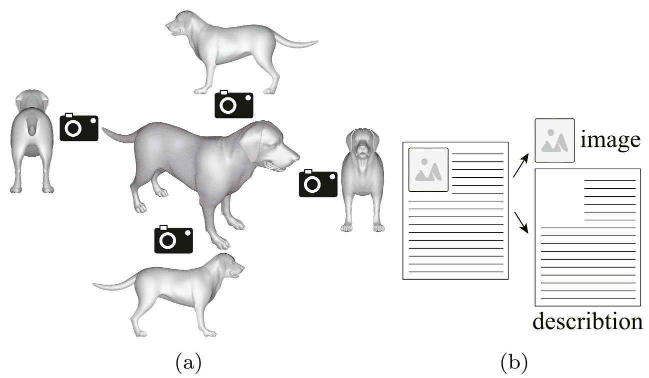


Fig. 1. Examples of multi-view data: (a) photos of the same dog taken from different viewpoints; (b) web image and its description which both completely describe an item. Each one of these data is regarded as a complementary view.

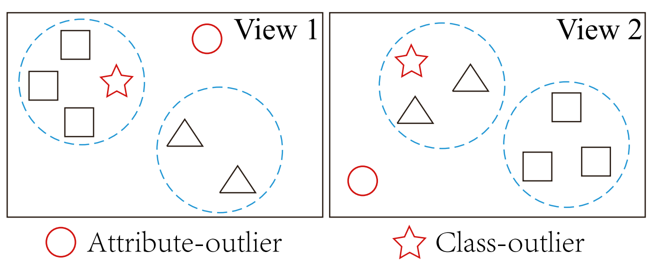


Fig. 2. Two types of multi-view outliers in two-view scenario: The *attribute-outlier* (circle) is remote from the other points in both views; The *class-outlier* (star) is classified as the square in the first view but as the triangle in the second view.

methods are easily processed in a parallel programming framework (e.g. MapReduce [17]), which optimizes the algorithms for distributed computational environment [18]. Many multi-view k-means methods [16,19–23] bring these advantages into various areas such as image segmentation [24,25], text clustering [26], document image analysis [27], color quantization [28], collaborative filtering [29], etc. There are two commonly used techniques to satisfy the robustness of clustering. One is using the specific norm to *weaken* the influence of outliers [16,30] and the other is using the outliers *removal* strategy to eliminate it [12,31]. Using the specific norm is the commonly used technique because of its simpleness, and using the outliers removal strategy seems to be more well-performing.

Different from the single view scenarios, multi-view outliers are a lot worth of study and even the definition of outliers has been discussed for a long term. In single-view data, the outlier is defined as a data point that deviates too much from other data points [32]. But in multi-view data, such a definition may no longer be appropriate, because an instance may exhibit inconsistent behavior across different views [33–35], i.e., the instance is assigned to different clusters for different views. Following [33, 36,37], two types of outliers for multi-view data are considered in this paper: (1) *Attribute-outliers* are those far away from others in most views, like the red circle in Fig. 2. (2) *Class-outliers* are those showing inconsistent characteristics, like the red star in Fig. 2.

However, most of the robust multi-view clustering methods only *weaken* the influence of *Attribute-outliers* by using the $L_{2,1}$-norm mentioned below, and no multi-view clustering methods can remove *class-outliers*, so they naturally have disadvantages in the dataset with larger noise ratio and more complex outlier form. In this paper, we propose a novel robust multi-view k-means method called Multi-View Clustering with Outlier Removal (MVCOR), which utilizes the heterogeneous representations of multi-view data by ignoring the inconsistency noises and fully using the latent consistency. For robust clustering, a well-designed outlier removal strategy is used to make MVCOR more robust to outliers in multi-view data and even the extreme outliers (the outliers lie farther away than the common outliers). More specifically, MVCOR categorizes all the two types of multi-view outliers into two additional clusters respectively, which significantly *removes* the interference of outliers and thus boosts the clustering performance on multi-view data. The main contributions of this paper are summarized as follows:

- MVCOR *simultaneously* performs clustering and detects different types of outliers, which achieves a more superior and stable clustering performance.
- MVCOR *removes* the negative influence of outliers in multi-view data, thereby further enhancing the clustering robustness and performance.
- MVCOR *scales almost linearly* with the data size, thus it can be easily applied to the large-scale data.

The rest of this paper is organized as follows. In Section 2, we give a brief review of robust multi-view clustering and outlier detection. In Section 3, we discuss how to find the two types of outliers in detail. We present our model in Section 4 and optimize the model in Section 5. The ability of clustering and finding outliers as well as the robustness of MVCOR will be tested in Section 6. Section 7 is a short conclusion.

2. Related work

2.1. Robust multi-view clustering

Multi-view clustering assumes that data from different views are consistent with an underlying clustering structure [22]. The well-known boosting techniques include co-training [38] and co-regularization [39], which both improve the performance of one view through other views. Another commonly-used multi-view technique is adaptive view weight updates [40]. In robust multi-view clustering, one of the prominent works is the $L_{2,1}$-norm, which is also known as sparsity-inducing norm [41]. In particular, the $L_{2,1}$-norm of matrix is defined as $\|X\|_{2,1} = \sum_i \|X^i\|_2$, where X^i is the ith row vector of the matrix X. Since the classical k-means algorithm uses the Frobenius norm and calculates the square distance between data points and their cluster centroids, *attribute-outliers* generate a large value in the objective function under this case. In contrast, the $L_{2,1}$-norm eliminates the exponent, which calculates distance but not square distance, and successfully weakens the outliers' contribution to the objective function. Many works introduce the $L_{2,1}$-norm. Kong et al. [42] adopt the $L_{2,1}$-norm into Non-negative Matrix Factorization (NMF). Cai et al. [16] adopt k-means into large-scale multi-view data by introducing the $L_{2,1}$-norm. Liang et al. [43] then extend it to the kernel space. Kannan et al. [44] propose an NMF method and used the $L_{2,1}$-norm to model the outliers. Jiang et al. [45] propose a new Vector Outlier Regularization (VOR) framework to understand and analyze the robustness of the $L_{2,1}$-norm function. However, the $L_{2,1}$-norm just weakens but not eliminates the influence of the outliers, and is still not robust enough in practice. The extreme outliers can easily make these methods invalid. But only a few works try to remove the negative influence of outliers. Gao et al. [46] introduce capped-norm, limiting the residual of outliers by setting a threshold, which is equivalent to removing the extreme values in each iteration. Huang et al. [12] then adopt it into multi-view k-means. Nevertheless, all these mentioned methods can only handle the *attribute-outliers* and neglect the *class-outliers*.

2.2. Outlier detection

Outlier detection aims to detect abnormal data in a given dataset [47]. In other words, the main goal of outlier detection is to find outliers or noises that are markedly different from or inconsistent with the normal instances [48]. Up to now, many single-view outlier detection methods have been proposed, including distance-based methods [49–51], density-based methods [52], etc. These methods have their advantages in different distributed datasets. When the data distribution is different from the expected assumptions, they often have very poor performance. In addition, some other research try to combine outlier detection and clustering, but they also mainly concentrate on single-view algorithm, such as ORC [53], k-means– [54], ODC [47], KMOR [31] and COR [55]. These methods try to find the outlier through analyzing the hidden clustering structure, so they cannot even work in non-clustering data. But their detecting results are more suitable to boost the corresponding clustering method than the non-clustering outlier detection methods. There are only a few multi-view outlier detection methods in the literature. The reason is that it is difficult to define multi-view outliers due to the increase in views. Gao et al. [34] firstly study the *inconsistent behavior* and propose a graph-based approach HOAD, which uses spectral technique and computes the *class-outlier* score by considering the cosine distance between the clustering result in different views. But this framework is too rough and only can find multi-view outliers preliminarily. Janeja et al. [56] raise a similar problem of multi-domain anomaly detection, and there is a preliminary consensus on the definition of multi-view outliers. Alvarez et al. [35] consider the affinity propagation algorithm and propose four types of anomaly score computation based on the clustering-based affinity vectors, which makes a great success and is often regarded as the baseline of multi-view outlier detection. Different from robust multi-view clustering, the aforementioned multi-view outlier detection algorithms mainly focus on detecting the *class-outlier*.

3. Preliminaries

Suppose there are N independent and identically distributed (i.i.d) instances and M different views of features, the aim is to partition these instances into K disjoint clusters. Each view has its representation of these N instances such that: $X^{(v)} = [X_1^{(v)}, \dots, X_n^{(v)}] \in \mathbb{R}^{\dim(v)\times \mathrm{N}}$, where $\dim(v)$ is the feature dimension in the vth view. Similarly, the cluster centroids in the vth view are denoted as $F^{(v)} = [F_1^{(v)}, F_2^{(v)}, \dots, F_K^{(v)}] \in \mathbb{R}^{\dim(v)\times \mathrm{K}}$. In this paper, the ith column of a matrix A is denoted as A_i (a column vector), the ith row of a matrix A is denoted as A^i (a row vector), and the element in the ith row, the jth column of a matrix A is denoted as $a_{i,j}$. The notations used throughout this paper are summarized in Table 1.

Table 1
Summary of notations.

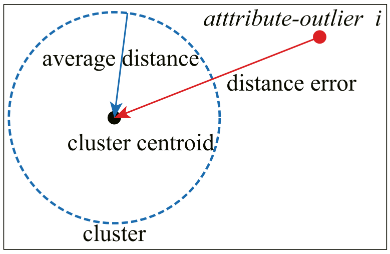
Notation	Description
X	Multi-view data set
N	The number of instances in X
M	The number of views
K	The number of clusters
F	Multi-view cluster centroids
$\dim(v)$	The number of features in the vth view
$g_{v,i}$	The index of the nearest cluster of the ith instance in the vth view
U	Cluster indicator matrix with the two types of outliers, i.e., $\mathrm{K}+1$ represents the *attribute-outlier* and $\mathrm{K}+2$ represents the *class-outlier*
$\overline{U}$	Cluster indicator matrix for K clusters without discriminate between the normal instances and the outliers
$\|\cdot\|_{2,1}$	the $\mathrm{L}_{2,1}$-norm
$\mathbb{1}[\cdot]$	The indicator function, and it equals to 1 if condition is true, otherwise equals to 0
Z^i	The ith row of the matrix Z
Z_i	The ith column of the matrix Z
$z_{i,j}$	The element in the ith row and the jth column of the matrix Z
$D^{(v)}$	The matrix corresponds to the vth view and the element corresponds to the ith instance and the kth centroid is defined as $d_{i,k}^{(v)} = (2\|X_i^{(v)} - F_k^{(v)}\|_2)^{-1}$
$\boldsymbol{\alpha}$	View weight factor vector
γ	Parameter to control the weight distribution
θ_1	Parameter to control the threshold between normal instance and *attribute-outlier*
θ_2	Parameter to control the threshold between normal instance and *class-outlier*
n_{max}	Parameter to control the maximum number of outliers the algorithm can find
$\mathbf{e}_i$	Row vector of adaptive length that the ith element is 1, and others are 0
A	The entirety of variable U, G, $\overline{U}$ and $\boldsymbol{\alpha}$
$a_{i,k}^{(v)}$	The sum of all coefficients of $d_{i,k}^{(v)}\|X_i^{(v)} - F_k^{(v)}\|_2^2$ in objective function Eq. (11)
$DV(i,k,v)$	The distance vector defined as $DV(i,k,v) = X_i^{(v)} - F_k^{(v)}$

3.1. Robust multi-view k-means clustering using $L_{2,1}$-norm (RMKMC)

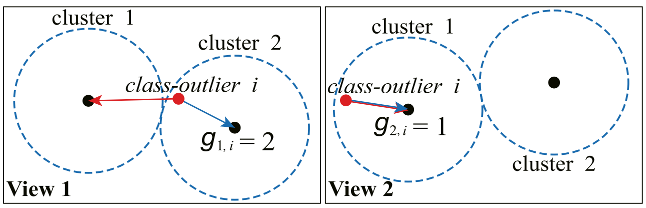
Multi-view k-means forces each instance to be assigned to the same cluster across all views. Thus, a common cluster indicator matrix U is shared among all views, in which $u_{i,k} = 1$ indicates the ith instance to be assigned to the kth cluster, otherwise $u_{i,k} = 0$. Each instance can only be assigned to one cluster, whose centroids in all views are nearest to it. Now, we introduce the general robust multi-view k-means using the $\mathrm{L}_{2,1}$-norm [16]:

$$
\begin{aligned}
&\min_{U,F,\boldsymbol{\alpha}} \sum_{v=1}^{M} (\boldsymbol{\alpha}_v)^{\gamma} \sum_{i=1}^{\mathrm{N}} \sum_{k=1}^{\mathrm{K}} u_{i,k} \|X_i^{(v)} - F_k^{(v)}\|_2 \\
&s.t. \quad \sum_{k=1}^{\mathrm{K}} u_{i,k} = 1,\ u_{i,k} \in \{0,1\},\ \forall i = 1,\dots,\mathrm{N}, \\
&\qquad \boldsymbol{\alpha}^T \mathbf{1} = 1,
\end{aligned}
\tag{1}
$$

where $(\boldsymbol{\alpha}_v)^{\gamma}$ is the weight of the vth view, γ is the parameter to control the weight distribution ($\gamma \to \infty$ gives all view the same weight, $\gamma = 1$ assigns 1 to the weight of the view whose residual is smallest and assigns 0 to the weights of other views). The $L_{2,1}$-norm minimization of the objective function is adopted to ensure the robustness to the outliers in data points. In this objective function, $u_{i,k} = 1$ when distance error $\sum_{v=1}^{M} (\boldsymbol{\alpha}_v)^{\gamma} \|X_i^{(v)} - F_k^{(v)}\|_2$ is minimal.



(a) Attribute-outlier



(b) Class-outlier

Fig. 3. An illustration of the presupposition in two view scenario: (a) The *attribute-outlier* satisfy that its distance error (the red arrow) is no less than some multiple of the average distance error (the blue arrow); (b) The *class-outlier* satisfy that its distance error in two views (sum of the two red arrows) is no less than some multiple of the distance error to the closest centroid respectively in two views (sum of the blue arrows). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. The outlier removal strategy

As mentioned earlier, there are two types of outliers in multi-view data: *class-outliers* and *attribute-outliers*. Because the k-means algorithm is based on distance, it is reasonable to use the distance-based method to model the *class-outlier* and the *attribute-outlier*. Distance error of the ith instance be assigned to the k_ith cluster is defined as:

$$DE(i) = \sum_{v=1}^{\mathrm{M}} (\boldsymbol{\alpha}_v)^{\gamma} \|X_i^{(v)} - F_{k_i}^{(v)}\|_2, \tag{2}$$

where $\boldsymbol{\alpha}_v$ and γ are defined in Section 3.1. The outlier removal strategy regards an instance as an outlier when its distance error is greater than a well-defined threshold.

For the *attribute-outlier*, one possible presupposition is that *attribute-outliers* always lay far away from its cluster centroids and its distance error is significantly greater than the average. As shown in Fig. 3(a), we suppose the threshold of judging *attribute-outlier* is defined as some multiple of the average distance error (assume that the ith instance is an *attribute-outlier*):

$$DE(i) > \theta_1 \underset{j=1,\ldots,\mathrm{N}}{avg} [DE(j)], \tag{3}$$

where θ_1 is the parameter, avg is the average function that calculates the average distance error.

The *class-outlier* performs *inconsistent behavior* and belongs to different clusters in different views. If we figured out *class-outliers*' closest cluster centroids in each view, the closest cluster centroids in some views were not the assigned cluster centroids and the distance error is greater than the sum of the single-view distance error to these closest cluster centroids. As shown in Fig. 3(b), the threshold of judging *class-outlier* is defined as some multiple of this sum (assuming that instance i is a *class-outlier*):

$$DE(i) > \theta_2 \sum_{v=1}^{\mathrm{M}} (\boldsymbol{\alpha}_v)^{\gamma} \|X_i^{(v)} - F_{g_{v,i}}^{(v)}\|_2, \tag{4}$$

where $g_{v,i}$ is the ith instance's closest cluster index in the vth view and θ_2 is the parameter. Normal instances have consistent behavior, thus their closest clusters in each view are the same, and the In Eq. (4) will take the mark of equality when $\theta_2 = 1$. However, *class-outliers* have different $g_{v,i}$ correspond to different v. Once we assume that the *class-outlier* performs consistent behavior and classifies it into the k_ith cluster, the left side of the In Eq. (4) will be greater since many of $g_{v,i}$ are not equal to k_i. An instance may be both *attribute-outlier* and *class-outlier*, but the outlier detection task is just to find outliers. Thus, the proposed algorithm is effective as long as the instance is classified as any kind of multi-view outliers.

4. Proposed method

To remove the negative influence of the two types of outliers, our method MVCOR combines both the $\mathrm{L}_{2,1}$-norm and the idea that separately classify outliers into specific clusters. We apply In Eq. (3) and In Eq. (4) as the criterion of judging *attribute-outliers* and *class-outliers*, classifying them into the cluster $\mathrm{K}+1$ and the cluster $\mathrm{K}+2$ respectively. Thus the objective function is further formulated as:

$$\begin{aligned}
\min_{U,F,G,\overline{U},\boldsymbol{\alpha}} & \sum_{i=1}^{\mathrm{N}} \Big(\sum_{k=1}^{\mathrm{K}} u_{i,k} \sum_{v=1}^{\mathrm{M}} (\boldsymbol{\alpha}_v)^{\gamma} \|X_i^{(v)} - F_k^{(v)}\|_2 \\
& + u_{i,\mathrm{K}+1} \mathrm{T}_1(\overline{U}, F) + u_{i,\mathrm{K}+2} \mathrm{T}_2(G_i, F) \Big) \\
s.t. \quad & \sum_{k=1}^{\mathrm{K}+2} u_{i,k} = 1,\, u_{i,k} \in \{0,1\},\, \forall i = 1,\ldots,\mathrm{N}, \\
& \sum_{k=1}^{\mathrm{K}} \overline{u}_{i,k} = 1,\, \overline{u}_{i,k} \in \{0,1\},\, \forall i = 1,\ldots,\mathrm{N}, \\
& \sum_{j=1}^{\mathrm{N}} (u_{j,\mathrm{K}+1} + u_{j,\mathrm{K}+2}) \le n_{max} \\
& g_{v,i} \in \{1,\ldots,\mathrm{N}\},\, \boldsymbol{\alpha}^T \mathbf{1} = 1,
\end{aligned} \tag{5}$$

where

$$\mathrm{T}_1(\overline{U}, F) = \theta_1 \underset{j=1,\ldots,\mathrm{N}}{avg} [DE(j)], \tag{6}$$

$$\mathrm{T}_2(G_i, F) = \theta_2 \sum_{v=1}^{\mathrm{M}} (\boldsymbol{\alpha}_v)^{\gamma} \|X_i^{(v)} - F_{g_{v,i}}^{(v)}\|_2, \tag{7}$$

and the average distance is formulated as:

$$\underset{j=1,\ldots,\mathrm{N}}{avg} [DE(j)] = \frac{1}{\mathrm{N}} \sum_{j=1}^{\mathrm{N}} \sum_{k=1}^{\mathrm{K}} \overline{u}_{j,k} \sum_{v=1}^{\mathrm{M}} (\boldsymbol{\alpha}_v)^{\gamma} \|X_j^{(v)} - F_k^{(v)}\|_2, \tag{8}$$

where X, F, G, $\boldsymbol{\alpha}_v$, γ, θ_1 and θ_2 are mentioned above. n_{max} limits the maximum number of outliers the algorithm can find and guarantees that it does not discard too many instances. In this place, $U \in \mathbb{R}^{\mathrm{N}\times(\mathrm{K}+2)}$ is the cluster indicator matrix for $\mathrm{K}+2$ clusters, i.e., $u_{i,\mathrm{K}+1} = 1$ when instance i is classified into the cluster $\mathrm{K}+1$ and is considered as an *attribute-outlier*, and $u_{i,\mathrm{K}+2} = 1$ represents the instance i is a *class-outlier*. $\overline{U} \in \mathbb{R}^{\mathrm{N}\times\mathrm{K}}$ is the cluster indicator matrix for K clusters if the proposed method perform clustering without judging *attribute-outliers* and *class-outliers*. From another perspective, $\overline{U}$ can be regarded as the classical clustering result. By removing *attribute-outliers* and *class-outliers*, we can get more robust clustering centroids, which improve the classical clustering result $\overline{U}$. $G \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}}$ records the index of nearest clusters for each instance and each view.

According to Eq. (5), when $u_{i,\mathrm{K}+1} = 1$, instance i is an *attribute-outlier* and it must satisfy the inequation:

$$DE(i) > \mathrm{T}_1(\overline{U}, F)\ \forall k = 1, \ldots, \mathrm{K}. \tag{9}$$

For the same reason, when $u_{i,\mathrm{K}+2} = 1$, instance i is an *class-outlier* and it must satisfy the inequation:

$$DE(i) > \mathrm{T}_2(G_i, F)\ \forall k = 1, \ldots, \mathrm{K}. \tag{10}$$

5. Optimization and analysis

Similar to the classical k-means clustering optimization, the efficient alternating minimization algorithm is used to optimize the Eq. (5). Alternating minimization algorithm solves optimization problems over more than one variables, each alternating step optimizes one variable and is usually convex and tractable. However, it is not easy to utilize the derivative method to the sum of a series of vector l2-norm. Firstly, we rewrite Eq. (5) by replacing all $\|X_i^{(v)} - F_k^{(v)}\|_2$ with $d_{i,k}^{(v)}\|X_i^{(v)} - F_k^{(v)}\|_2^2$:

$$\begin{aligned}
\min_{U,F,G,\overline{U},D,\boldsymbol{\alpha}} & \sum_{i=1}^{\mathrm{N}}\Big(\sum_{k=1}^{\mathrm{K}} u_{i,k}\sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma} d_{i,k}^{(v)}\|X_i^{(v)} - F_k^{(v)}\|_2^2 \\
& + u_{i,\mathrm{K}+1}\mathrm{T}_1(\overline{U}, F) + u_{i,\mathrm{K}+2}\mathrm{T}_2(G_i, F)\Big) \\
s.t.\quad & \sum_{k=1}^{\mathrm{K}+2} u_{i,k} = 1, u_{i,k} \in \{0,1\}, \forall i = 1, \ldots, \mathrm{N}, \\
& \sum_{k=1}^{\mathrm{K}} \overline{u}_{i,k} = 1, \overline{u}_{i,k} \in \{0,1\}, \forall i = 1, \ldots, \mathrm{N}, \\
& \sum_{j=1}^{\mathrm{N}} (u_{j,K+1} + u_{j,K+2}) \le n_{max} \\
& g_{v,i} \in \{1, \ldots, \mathrm{N}\}, \boldsymbol{\alpha}^T\mathbf{1} = 1,
\end{aligned} \tag{11}$$

where

$$\mathrm{T}_1(\overline{U}, F) = \frac{\theta_1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}}\sum_{k=1}^{\mathrm{K}}\overline{u}_{i,k}\sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma} d_{i,k}^{(v)}\|X_i^{(v)} - F_k^{(v)}\|_2^2, \tag{12}$$

$$\mathrm{T}_2(G_i, F) = \theta_2\sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma} d_{i,g_{v,i}}^{(v)}\|X_i^{(v)} - F_{g_{v,i}}^{(v)}\|_2^2, \tag{13}$$

$$d_{i,k}^{(v)} = \frac{1}{2\|X_i^{(v)} - F_k^{(v)}\|_2}. \tag{14}$$

We will illustrate that we can get the same result of the origin objective function Eq. (5) by solving the rewritten objective function Eq. (11). And the alternating minimization algorithm of optimizing Eq. (11) is also convergent.

5.1. Fix U, G, $\overline{U}$, $\boldsymbol{\alpha}$, D and update F

For view v and cluster k, optimizing Eq. (11) w.r.t $F_k^{(v)}$ is equivalent to minimizing the problem as follows:

$$\begin{aligned}
J = & \arg\min_f \sum_{i=1}^{\mathrm{N}}(u_{i,k}d_{i,k}^{(v)}\|X_i^{(v)} - f\|_2^2 \\
& + u_{i,\mathrm{K}+1}\frac{\theta_1}{\mathrm{N}}\sum_{j=1}^{\mathrm{N}}\overline{u}_{j,k}d_{j,k}^{(v)}\|X_j^{(v)} - f\|_2^2 \\
& + u_{i,\mathrm{K}+2}\theta_2 d_{i,k}^{(v)}\|X_i^{(v)} - f\|_2^2\mathbb{1}[g_{v,i} = k]),
\end{aligned} \tag{15}$$

where $\mathbb{1}[g_{v,i} = k]$ is the indicator function, and it equals to 1 if $g_{v,i} = k$, otherwise equals to 0. Taking the derivative of Eq. (15) w.r.t. f, we have:

$$\begin{aligned}
\frac{\partial J}{\partial f} = & \sum_{i=1}^{\mathrm{N}}(2u_{i,k}d_{i,k}^{(v)}(f - X_i^{(v)}) \\
& + 2u_{i,\mathrm{K}+1}\frac{\theta_1}{\mathrm{N}}\sum_{j=1}^{\mathrm{N}}\overline{u}_{j,k}d_{j,k}^{(v)}(f - X_j^{(v)}) \\
& + 2u_{i,\mathrm{K}+2}\theta_2 d_{i,k}^{(v)}(f - X_i^{(v)})\mathbb{1}[g_{v,i} = k]).
\end{aligned} \tag{16}$$

Setting Eq. (16) to 0, we get the updating rule of $F_k^{(v)}$ given in Box I.

5.2. Fix U, F, G, $\overline{U}$, $\boldsymbol{\alpha}$ and update D

We update D by Eq. (14).

5.3. Fix U, F, $\boldsymbol{\alpha}$, D and update G, $\overline{U}$

$g_{v,i}$ is calculated by its definition:

$$g_{v,i} = \arg\min_k \|X_i^{(v)} - F_k^{(v)}\|_2^2. \tag{18}$$

For $\overline{U}$, we have:

$$\min_{\overline{U}} \sum_{i=1}^{\mathrm{N}}\sum_{k=1}^{\mathrm{K}}\overline{u}_{i,k}\sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma} d_{i,k}^{(v)}\|X_i^{(v)} - F_k^{(v)}\|_2^2. \tag{19}$$

To minimize the Eq. (19), we tackle it for each instance one by one. For a specific instance i, we find the solution by searching all possible answer:

$$\overline{U}^i = \mathbf{e}_{\overline{k}}, \tag{20}$$

in which

$$\overline{k} = \underset{k}{argmin}\sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma} d_{i,k}^{(v)}\|X_i^{(v)} - F_k^{(v)}\|_2^2, \tag{21}$$

where $\mathbf{e}_i$ is a row vector that the ith element is 1, and others are 0.

5.4. Fix $\boldsymbol{F}$, $\boldsymbol{G}$, $\overline{\boldsymbol{U}}$, $\boldsymbol{\alpha}$, $\boldsymbol{D}$ and update $\boldsymbol{U}$

We also use the exhaustive search strategy to find the solution:

$$U^i = \mathbf{e}_k, \tag{22}$$

in which

$$k = \begin{cases} \overline{k}, & \overline{U}_{i,\overline{k}} = 1, \\ & \sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma} d_{i,\overline{k}}^{(v)}\|X_i^{(v)} - F_{\overline{k}}^{(v)}\|_2^2 \text{ is minimum} \\ \mathrm{K}+1, & \mathrm{T}_1(\overline{U}, F) \text{ in Eq. (12) is minimum} \\ \mathrm{K}+2, & \mathrm{T}_2(G_i, F) \text{in Eq. (13) is minimum.} \end{cases} \tag{23}$$

However, this may not satisfy the constraint $\sum_{j=1}^{\mathrm{N}}(u_{j,K+1}+u_{j,K+2}) \le n_{max}$. When there is more than n_{max} outlier in this iteration, we maintain the top n_{max} outliers according to the difference between the distance error and $\mathrm{T}_1(\overline{U}, F)$ (when $k = K+1$ in Eq. (23)) or $\mathrm{T}_2(G_i, F)$ (when $k = K+2$ in Eq. (23)), i.e., let

$$\begin{aligned}
& i_1, i_2, \ldots, i_{n_{max}} \\
= & \arg\max_{1\le i\le n_{max}}\{\sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma} d_{i,\overline{k}}^{(v)}\|X_i^{(v)} - F_{\overline{k}}^{(v)}\|_2^2 \\
& - \mathbb{1}[k = K+1]\mathrm{T}_1(\overline{U}, F) - \mathbb{1}[k = K+2]\mathrm{T}_2(G_i, F)\},
\end{aligned} \tag{24}$$

$$F_k^{(v)}=\frac{\sum_{i=1}^{\mathrm{N}}(u_{i,k}d_{i,k}^{(v)}X_i^{(v)}+u_{i,\mathrm{K}+1}\frac{\theta_1}{\mathrm{N}}\sum_{j=1}^{\mathrm{N}}\overline{u}_{j,k}d_{j,k}^{(v)}X_j^{(v)}+u_{i,\mathrm{K}+2}\theta_2 d_{i,k}^{(v)}X_i^{(v)}\mathbb{1}[g_{v,i}=k])}{\sum_{i=1}^{\mathrm{N}}(u_{i,k}d_{i,k}^{(v)}+u_{i,\mathrm{K}+1}\frac{\theta_1}{\mathrm{N}}\sum_{j=1}^{\mathrm{N}}\overline{u}_{j,k}d_{j,k}^{(v)}+u_{i,\mathrm{K}+2}\theta_2 d_{i,k}^{(v)}\mathbb{1}[g_{v,i}=k])}. \tag{17}$$

Box I.

where $\bar{k}$ satisfies that $\overline{U}_{i,\bar{k}}=1$ and k is derived by Eq. (23). The outliers which are not in $\{i_1, i_2, \ldots, i_{n_{max}}\}$ are reset by $\overline{U}$:

$$U^i=\overline{U}^i,\ \forall i \text{ not in } \{i_1, i_2, \ldots, i_{n_{max}}\}. \tag{25}$$

5.5. Fix U, F, G, $\overline{U}$, D and update $\boldsymbol{\alpha}$

Optimizing Eq. (11) w.r.t. $\boldsymbol{\alpha}_v$ is equivalent to minimizing the problem as follows:

$$\begin{aligned}\min_{U,F,G,\overline{U},D,\boldsymbol{\alpha}} &\sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma}\sum_{i=1}^{\mathrm{N}}\Big(\sum_{k=1}^{\mathrm{K}}u_{i,k}d_{i,k}^{(v)}\|X_i^{(v)}-F_k^{(v)}\|_2^2\\ &+u_{i,\mathrm{K}+1}\frac{\theta_1}{\mathrm{N}}\sum_{j=1}^{\mathrm{N}}\sum_{k=1}^{\mathrm{K}}\overline{u}_{j,k}d_{j,k}^{(v)}\|X_j^{(v)}-F_k^{(v)}\|_2^2\\ &+u_{i,\mathrm{K}+2}\theta_2 d_{i,g_{v,i}}^{(v)}\|X_i^{(v)}-F_{g_{v,i}}^{(v)}\|_2^2\Big).\end{aligned} \tag{26}$$

Let

$$\begin{aligned}\mathrm{H}(v)=&\sum_{i=1}^{\mathrm{N}}\Big(\sum_{k=1}^{\mathrm{K}}u_{i,k}d_{i,k}^{(v)}\|X_i^{(v)}-F_k^{(v)}\|_2^2\\ &+u_{i,\mathrm{K}+1}\frac{\theta_1}{\mathrm{N}}\sum_{j=1}^{\mathrm{N}}\sum_{k=1}^{\mathrm{K}}\overline{u}_{j,k}d_{j,k}^{(v)}\|X_j^{(v)}-F_k^{(v)}\|_2^2\\ &+u_{i,\mathrm{K}+2}\theta_2 d_{i,g_{v,i}}^{(v)}\|X_i^{(v)}-F_{g_{v,i}}^{(v)}\|_2^2\Big),\end{aligned} \tag{27}$$

and the Eq. (26) w.r.t. $\boldsymbol{\alpha}$ is simplified as:

$$\begin{aligned}&\min_{\boldsymbol{\alpha}}\sum_{v=1}^{\mathrm{M}}(\boldsymbol{\alpha}_v)^{\gamma}\mathrm{H}(v)\\ &s.t.\quad \boldsymbol{\alpha}^T\mathbf{1}=1.\end{aligned} \tag{28}$$

Taking Lagrangian function of Eq. (28) and solving it, we have:

$$\boldsymbol{\alpha}_v=\frac{(\gamma\mathrm{H}(v))^{\frac{1}{1-\gamma}}}{\sum_{\bar{v}=1}^{\mathrm{M}}(\gamma\mathrm{H}(\bar{v}))^{\frac{1}{1-\gamma}}}. \tag{29}$$

Notice that $\mathrm{H}(v)\geq 0$, $\gamma\in(1,+\infty)$, which ensures $\boldsymbol{\alpha}_v\geq 0$.

5.6. Convergence analysis

In this section, we provide the convergence analysis in Algorithm 1.

Theorem 1. *Solving the rewritten objective function Eq. (11) is equivalent to solving the objective function Eq. (5).*

Proof. The rewritten objective function Eq. (11) and the objective function Eq. (5) are represented by $\mathrm{obj_{new}}$ and $\mathrm{obj_{old}}$ respectively. In tth iteration, we have

$$\mathrm{obj_{new}}(U^t,F^t,G^t,\overline{U},\boldsymbol{\alpha}^t,D^t)=\mathrm{obj_{old}}(U^t,F^t,G^t,\overline{U},\boldsymbol{\alpha}^t).\quad\square \tag{30}$$

Before introducing the theoretical analysis w.r.t. Algorithm 1, we start by a significant Lemma in [30]:

Algorithm 1 The algorithm of MVCOR

Input:
Data for M views $\{X^{(1)},\ldots,X^{(\mathrm{M})}\}$ and $X^{(v)}\in\mathbb{R}^{\dim(v)\times\mathrm{N}}$;
The expected number of clusters K;
The parameters θ_1, θ_2, γ and n_{max}.
Output:
The cluster assignment matrix U with the two types of outliers;
The cluster assignment matrix $\overline{U}$ without the two types of outliers;
The cluster centroid matrix $F^{(v)}$ for each view.
Initialization:
Initialize $t=0$, $\boldsymbol{\alpha}_v=\frac{1}{\mathrm{M}}$, $d_{i,k}^{(v)}=1$;
Initialize U and $\overline{U}$ with the same values.
repeat
  Update $F_k^{(v)}$ by Eq. (17);
  Update D by Eq. (14);
  Update $g_{v,i}$ by Eq. (18), $\overline{U}^i$ by Eqs. (20) and (21);
  Update U^i by Eqs. (22) and (23), and if there is more than n_{max} outlier, reset some outliers by Eqs. (24) and (25);
  Update $\boldsymbol{\alpha}_v$ by Eqs. (27) and (29);
  $t=t+1$.
until Converges

Lemma 1.

$$\|a\|_2-\|b\|_2\leq\frac{\|a\|_2^2}{2\|b\|_2}-\frac{\|b\|_2^2}{2\|b\|_2}\ \forall a,b\in\mathbb{R}^c,\ a\neq\mathbf{0},\ b\neq\mathbf{0}. \tag{31}$$

Proof.

$$\begin{aligned}&(\|b\|_2-\|a\|_2)^2\geq 0\\ \Rightarrow&\|b\|_2^2-2\|b\|_2\|a\|_2+\|a\|_2^2\geq 0\\ \Rightarrow&2\|b\|_2\|a\|_2-\|a\|_2^2\leq 2\|b\|_2^2-\|b\|_2^2\\ \Rightarrow&\|a\|_2-\frac{\|a\|_2^2}{2\|b\|_2}\leq\|b\|_2-\frac{\|b\|_2^2}{2\|b\|_2}\\ \Rightarrow&\|a\|_2-\|b\|_2\leq\frac{\|a\|_2^2}{2\|b\|_2}-\frac{\|b\|_2^2}{2\|b\|_2}\quad\square\end{aligned} \tag{32}$$

Theorem 2. *Even if introducing matrix D, our alternating minimization algorithm also decrease the objective function Eq. (11) in each iteration.*

Proof. Firstly, we regard the variable U, G, $\overline{U}$ and $\boldsymbol{\alpha}$ as an entirety, represented by the upper letter A. The target is to prove

$$\mathrm{obj_{new}}(D^t,F^t,\mathrm{A}^t)\geq\mathrm{obj_{new}}(D^{t+1},F^{t+1},\mathrm{A}^t) \tag{33}$$

and

$$\mathrm{obj_{new}}(D^{t+1},F^{t+1},\mathrm{A}^t)\geq\mathrm{obj_{new}}(D^{t+1},F^{t+1},\mathrm{A}^{t+1}). \tag{34}$$

Notice that all the steps in Algorithm 1 decrease the objective function Eq. (11) except for updating D, so In Eq. (34) is always true. To prove In Eq. (33), we reformulate the objective function Eq. (11):

$$\min_{A,D,F} \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)} d_{i,k}^{(v)} \|DV(i,k,v)\|_2^2, \tag{35}$$

where $DV(i,k,v) = X_i^{(v)} - F_k^{(v)}$, $a_{i,k}^{(v)}$ is the sum of all coefficients of $d_{i,k}^{(v)}\|X_i^{(v)} - F_k^{(v)}\|_2^2$. Notice that $a_{i,k}^{(v)}$ is irrelevant to F and D. According to Eqs. (35) and (14), we have:

$$\begin{aligned}
&\mathrm{obj}_{new}(A^t, D^{t+1}, F^{t+1}) - \mathrm{obj}_{new}(A^t, D^t, F^t)\\
=&\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} d_{i,k}^{(v)^{t+1}} \|DV(i,k,v)^{t+1}\|_2^2 - \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} d_{i,k}^{(v)^t} \|DV(i,k,v)^t\|_2^2\\
=&\frac{1}{2}\Big(\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} \|DV(i,k,v)^{t+1}\|_2 - \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} \|DV(i,k,v)^t\|_2\Big).
\end{aligned} \tag{36}$$

And then, the function can be scaled down by Lemma 1 In Eq. (31):

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} \|DV(i,k,v)^{t+1}\|_2 - \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} \|DV(i,k,v)^t\|_2\\
\le&\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} \frac{\|DV(i,k,v)^{t+1}\|_2^2}{2\|DV(i,k,v)^t\|_2} - \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} \frac{\|DV(i,k,v)^t\|_2^2}{2\|DV(i,k,v)^t\|_2}\\
=&\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} d_{i,k}^{(v)^t} \|DV(i,k,v)^{t+1}\|_2^2 - \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} d_{i,k}^{(v)^t} \|DV(i,k,v)^t\|_2^2.
\end{aligned} \tag{37}$$

Because the step of updating F is not increasing, we have

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} d_{i,k}^{(v)^t} \|DV(i,k,v)^{t+1}\|_2^2\\
&- \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{v=1}^{M} a_{i,k}^{(v)^t} d_{i,k}^{(v)^t} \|DV(i,k,v)^t\|_2^2 \le 0.
\end{aligned} \tag{38}$$

Thus we demonstrate the In Eq. (33). Both In Eq. (33) and In Eq. (34) are always true, so our algorithm is decreasing in each iteration. □

From Theorems 1 and 2, the alternating minimization algorithm cooperates to decrease both objective function Eqs. (5) and (11), and will converge to the suboptimal solution.

5.7. Discussion of the parameter θ_1 and θ_2

MVCOR involves four parameters: γ, θ_1, θ_2 and n_{max}. The parameter γ is to control the view weight distribution and the detail of γ is originated in [16]. The parameter n_{max} is to make sure that clustering can be carried out normally and MVCOR does not find too many outliers. The parameter θ_1 and θ_2 are the key parameters to find outliers. However, the appropriate value of these two parameters may greatly differ in various datasets. In this subsection, we focus on the method of setting parameters θ_1 and θ_2 by assuming the rates of *class-outliers* and *attribute-outliers*. We first run MVCOR with no outlier detection, i.e., setting a very large value to the parameter θ_1 and θ_2. And then, we calculate the smallest $\theta_1^{(i)}$ and $\theta_2^{(i)}$ for each instance i when the instance is *opportuney* classified as *class-outlier* and *attribute-outlier* respectively. Next, we sort the $\theta_1^{(i)}$ and $\theta_2^{(i)}$, and choose the suitable θ_1 and θ_2 from largest to smallest according to the assumed rate of *class-outliers* and *attribute-outliers*. The detail of this method is shown in Algorithm 2. This algorithm can work effectively because the clustering centroids of the non-robust multi-view clustering is not far away from its true position. In other words, outliers cannot cover useful information. Therefore, we can estimate the true distance and clusters' centroids, and further estimate the parameter θ_1 and θ_2.

Algorithm 2 The algorithm of getting appropriate parameters θ_1 and θ_2

Input:
Data for M views $\{X^{(1)}, \dots, X^{(M)}\}$ and $X^{(v)} \in \mathbb{R}^{\dim(v)\times N}$;
The expected number of clusters K;
The parameter γ;
The rates of *class-outliers* n_c and *attribute-outliers* n_a.
Output:
The parameters θ_1 and θ_2.
1: Run multiple times of general robust multi-view k-means using $L_{2,1}$-norm, or called RMKMC [16]. Choose the result when the value of the objective function Eq. (1) is minimal;
2: Calculate the distance error $DE(i)$ between the ith instance and its assigned cluster centroid;
3: Calculate the average distance $\underset{j=1,\dots,N}{avg}[DE(j)]$;
4: Calculate $\frac{DE(i)}{\underset{j=1,\dots,N}{avg}[DE(j)]}$, and select the $\lceil N \times n_a \rceil$th large value as θ_1;
5: Calculate $g_{v,i}$ by Eq. (18);
6: Calculate the distance error $DE'(i)$ between the ith instance and the cluster $g_{v,i}$;
7: Calculate $\frac{DE(i)}{DE'(i)}$, and select the $\lceil N \times n_c \rceil$th large value as θ_2.

5.8. Complexity analysis

All notations used below can be referred in Table 1, and $d_{max} = \max\{\dim(v)\}$. In each iteration, the cost of updating F by Eq. (17) is $O(NMKd_{max})$, the cost of updating D by Eq. (14) is $O(NMKd_{max})$, the cost of updating G by Eq. (18) is $O(NMKd_{max})$, the cost of updating $\overline{U}$ by Eqs. (20) and (21) is $O(NMKd_{max})$, the cost of calculating Eq. (12) is $O(NMKd_{max})$, the cost of calculating Eq. (13) is $O(NMd_{max})$, the cost of updating U by Eqs. (22) and (23) is $O(NMKd_{max})$, the reset step need to sort the value in Eq. (24), the average cost of quick sort is $O(N\log N)$, the cost of updating $\boldsymbol{\alpha}$ by Eqs. (27) and (29) is $O(NMKd_{max})$. To sum up, if the number of iterations is t, the overall time complexity of MVCOR is $O(tNMKd_{max} + tN\log N)$. The $N\log N$ term is only relative to Eq. (24), and will not be executed many times with appropriate parameters θ_1 and θ_2. Moreover, the number of outliers which need sorted will be much less than N in this case. Therefore, $\log N$ is much less than MKd_{max} in most practical case, and MVCOR cost a nearly linear time complexity with the data size.

6. Experiments

Because there is no algorithm for both multi-view clustering and multi-view outlier detection, in this section, we evaluate MVCOR in three-part: comparing its clustering performance, analyzing its outlier detection ability, and illustrating its robustness.

6.1. Clustering

6.1.1. Datasets

This experiment is performed on several real-world benchmark datasets:

Table 2
Description of multi-view datasets (with feature dimension in parenthesis)

View	SensIT-3000	MSRC-v1	AWA-10
1	Acoustic (50)	CM (24)	CQ (2688)
2	Seismic (50)	HOG (576)	LSS (2000)
3	–	GIST (512)	PHOG (252)
4	–	LBP (256)	RGSIFT (2000)
5	–	CENT (254)	SIFT (2000)
6	–	–	SURF (2000)
Number of instances	3000	210	800
Number of classes	3	7	10

- *SensIT Vehicle* [57] is a dataset from wireless distributed sensor networks, which consists of two views of data recorded by two different sensors. We download the dataset from LIBSVM and randomly sample 1000 instances for each class, regarded as a new dataset named *SensIT-3000*.
- *MSRC-v1* [58] is a scene recognition dataset from Microsoft Research in Cambridge. Following [2], we select 7 classes, i.e., tree, building, airplane, cow, face, car, bicycle, and use 5 visual feature extraction method to generate each view of data.
- *Animal With Attributes (AWA)* [59] is a large-scale dataset, which consists of six views of data. We select first 10 classes, i.e., antelope, bat, beaver, blue whale, bobcat, buffalo, chihuahua, chimpanzee, collie, cow, and randomly sample 80 instances for each class, regarded as a new dataset named *AWA-10*.

Table 2 summarizes the detailed information of the multi-view datasets.

6.1.2. Evaluation metrics

Six popular clustering evaluation metrics are used to measure the clustering performance, including Cluster Accuracy (ACC) [60], Normalized Mutual Information (NMI) [61], Purity [42], Adjusted Rand Index (ARI), Rand Index(RI) and F-Measure (FM). Specifically, these metrics range in [0, 1] (ACC, NMI, Purity, RI, FM) or [−1, 1] (ARI), and higher scores mean better clustering performance.

6.1.3. Comparison algorithms and experimental setup

Several multi-view clustering methods are employed to compare with MVCOR:

- *Co-train* [38] is the co-training multi-view spectral clustering, which bootstraps the spectral clustering results of different views using the eigenvalue matrices from one another.
- *Co-reg* [39] is the centroid-based co-regularized multi-view spectral clustering, which regularizes each view-specific eigenvalue matrices towards a common centroid matrix. Both co-training and co-regularization are well-known techniques in multi-view clustering which are set to compare with the view weight technique.
- *RMKMC* [16] is the robust multi-view k-means clustering using $L_{2,1}$-norm, which the logarithm of the parameter γ is searched in range [0.1, 2]. It is worth mentioning that RMKMC can be regarded as the fundamental version of our method without outlier removal strategy.
- *DEKM* [62] is the discriminatively embedded k-means, which is based on weighted multi-view *Linear Discriminant Analysis* (LDA) and can efficiently handle high dimensional data. This is the baseline of non-robust multi-view k-means clustering method.
- *CaMVC* [12] is the improved version of RMKMC in robustness by using capped-norm, which chooses the extreme data outliers by a thresholding parameter. But it can only handle *attribute-outliers*. Notice that *RMKMC*, *CaMVC* and our method are view-weight-based methods.

The classical single-view k-means algorithm is used as the baseline, and is applied on each view of datasets, e.g., KM (view 1) is the algorithm run on the 1st view as well as KM (all) is on the concatenated features of all views.

As for the parameter selection, we use the same γ of RMKMC, as well as run Algorithm 2 to search the appropriate θ_1 and θ_2 in the rate of 0% $\sim$ 9% respectively. The n_{max} is simply set to 0.2 number of instances. U and $\overline{U}$ are initialized randomly and satisfied that the instance numbers of different clusters are nearly equal. That is, we first generate an index vector of length N which is satisfied that the first $\lfloor N/K \rfloor$ elements are 1, the second $\lfloor N/K \rfloor$ elements are 2, . . . , the Kth $\lfloor N/K \rfloor$ elements are K, and the rest $N - \lfloor rmN/K \rfloor * K$ elements are random integer values range in [1, K]. And then we sort the index vector randomly. If the ith element of the index vector equals to k, $u_{i,k}$ and $\overline{u}_{i,k}$ is initialized to 1, otherwise is initialized to 0.

For MVCOR, the matrix $\overline{U}$ is regarded as the clustering result. We run each method for 30 times and show the mean values and standard deviations.

6.1.4. Experimental results

The experimental results are shown in Tables 3, 4, 5, and the best results in each dataset are highlighted in boldface. Table 6 shows the significant difference test in accuracy.

- Most multi-view clustering methods perform better than the baseline method in single-view data. Thus it is meaningful to use the integrated information to improve the clustering performance.
- Most multi-view clustering methods outperform the baseline method that directly concatenate all views of data. This is because the latter method cannot dig out the most useful information, but treat all views of data equally.
- Co-training and co-regularization continuous display excellent results, but other view-weight-based methods perform slightly behind them, illustrating the effectiveness of using view weight technique.
- RMKMC works better than DEKM, illustrating that robustness to outliers can boost the clustering performance.
- MVCOR works better than RMKMC, illustrating the superior of the outlier removal strategy and that the only use of $L_{2,1}$-norm is not enough to boost the clustering performance.
- MVCOR works better than CaMVC, and we log the outliers found by MVCOR in the SensIT-3000 dataset and find that the number of *class-outliers* is far more than the number of *attribute-outliers*. MVCOR performs better than CaMVC which can only tackle *attribute-outliers* in this dataset, showing that removing *class-outliers* improves the clustering performance to some extent.

6.2. Identification of outliers

6.2.1. Data generation

Following the data setting in [35] and [33], we select 7 UCI machine learning datasets: *Iris*, *Letter*, *Ionsphere*, *Zoo*, *Waveform*, *Pima*, and *Wdbc*. To save evaluation time, we randomly select some instances from each dataset, and Table 7 summarizes the detailed information. Since they are all single-view datasets, we partition the features into two views equally. To generate *class-outliers*, we randomly extract two instances from different classes and swap the features in one view and remain unchanged in

Table 3
Clustering results on SensIT-3000 dataset.

Methods	ACC	NMI	Purity	ARI	RI	FM
KM (view 1)	0.5387 ± 0.07	0.1370 ± 0.04	0.5432 ± 0.06	0.1378 ± 0.05	0.5795 ± 0.05	0.5577 ± 0.04
KM (view 2)	0.6445 ± 0.02	0.2738 ± 0.02	0.6445 ± 0.02	0.2873 ± 0.02	0.6773 ± 0.01	0.6410 ± 0.02
KM(all)	0.6671 ± 0.03	0.2912 ± 0.02	0.6671 ± 0.03	0.3147 ± 0.03	0.6900 ± 0.02	0.6609 ± 0.03
Co-train	0.6892 ± 0.04	0.3093 ± 0.03	0.6904 ± 0.04	0.3314 ± 0.04	$\mathbf{0.6982 \pm 0.02}$	$\mathbf{0.6875 \pm 0.03}$
Co-reg	0.6953 ± 0.00	$\mathbf{0.3327 \pm 0.00}$	0.6953 ± 0.00	0.3412 ± 0.06	0.6899 ± 0.34	0.6749 ± 0.23
RMKMC	0.6684 ± 0.03	0.3022 ± 0.02	0.6884 ± 0.03	0.3235 ± 0.03	0.6870 ± 0.02	0.6623 ± 0.02
DEKM	0.6671 ± 0.02	0.2941 ± 0.01	0.6671 ± 0.01	0.3164 ± 0.02	0.6895 ± 0.01	0.6574 ± 0.02
CaMVC	0.6738 ± 0.02	0.3042 ± 0.01	0.6738 ± 0.02	0.3229 ± 0.02	0.6889 ± 0.01	0.6615 ± 0.01
MVCOR	$\mathbf{0.6957 \pm 0.01}$	0.3217 ± 0.01	$\mathbf{0.6957 \pm 0.01}$	$\mathbf{0.3442 \pm 0.01}$	0.6912 ± 0.00	0.6716 ± 0.01

Table 4
Clustering results on MSRC-v1 dataset.

Methods	ACC	NMI	Purity	ARI	RI	FM
KM (view 1)	0.3438 ± 0.02	0.2331 ± 0.02	0.3733 ± 0.02	0.1127 ± 0.01	0.7800 ± 0.01	0.3626 ± 0.02
KM (view 2)	0.5641 ± 0.07	0.4694 ± 0.05	0.5813 ± 0.06	0.3513 ± 0.06	0.8359 ± 0.02	0.5836 ± 0.05
KM (view 3)	0.5719 ± 0.05	0.4802 ± 0.05	0.5962 ± 0.05	0.3766 ± 0.05	0.8395 ± 0.02	0.5872 ± 0.05
KM (view 4)	0.4827 ± 0.03	0.4256 ± 0.03	0.5181 ± 0.03	0.2667 ± 0.03	0.8090 ± 0.02	0.5141 ± 0.02
KM (view 5)	0.4641 ± 0.04	0.4058 ± 0.02	0.5065 ± 0.03	0.2465 ± 0.03	0.8031 ± 0.01	0.5015 ± 0.03
KM (all)	0.6813 ± 0.07	0.5849 ± 0.06	0.7056 ± 0.06	0.4984 ± 0.08	0.8729 ± 0.02	0.6895 ± 0.07
Co-train	0.7030 ± 0.04	0.6159 ± 0.03	0.7305 ± 0.04	0.5129 ± 0.05	0.8702 ± 0.02	0.7102 ± 0.04
Co-reg	0.5900 ± 0.02	0.5083 ± 0.02	0.5960 ± 0.02	0.4065 ± 0.02	0.8352 ± 0.01	0.6079 ± 0.01
RMKMC	0.7025 ± 0.08	0.6233 ± 0.07	0.7267 ± 0.07	0.5303 ± 0.09	0.8840 ± 0.02	0.7049 ± 0.07
DEKM	0.6346 ± 0.07	0.5471 ± 0.04	0.6540 ± 0.06	0.4346 ± 0.06	0.8585 ± 0.02	0.6476 ± 0.06
CaMVC	0.6975 ± 0.07	0.6162 ± 0.06	0.7267 ± 0.06	0.5259 ± 0.08	0.8833 ± 0.02	0.6982 ± 0.06
MVCOR	$\mathbf{0.7352 \pm 0.06}$	$\mathbf{0.6485 \pm 0.05}$	$\mathbf{0.7635 \pm 0.05}$	$\mathbf{0.5669 \pm 0.07}$	$\mathbf{0.8924 \pm 0.02}$	$\mathbf{0.7376 \pm 0.06}$

Table 5
Clustering results on AWA-10 dataset.

Methods	ACC	NMI	Purity	ARI	RI	FM
KM (view 1)	0.2070 ± 0.02	0.0970 ± 0.02	0.2246 ± 0.03	0.0360 ± 0.01	0.6409 ± 0.09	0.2455 ± 0.02
KM (view 2)	0.1942 ± 0.01	0.0842 ± 0.01	0.2083 ± 0.01	0.0301 ± 0.01	0.7046 ± 0.04	0.2196 ± 0.01
KM (view 3)	0.2137 ± 0.01	0.1021 ± 0.00	0.2241 ± 0.01	0.0428 ± 0.01	0.7185 ± 0.03	0.2530 ± 0.01
KM (view 4)	0.2311 ± 0.02	0.1047 ± 0.02	0.2447 ± 0.02	0.0557 ± 0.01	0.7718 ± 0.03	0.2499 ± 0.02
KM (view 5)	0.2435 ± 0.02	0.1425 ± 0.01	0.2609 ± 0.01	0.0687 ± 0.01	0.7886 ± 0.02	0.2710 ± 0.01
KM (view 6)	0.2566 ± 0.02	0.1386 ± 0.01	0.2701 ± 0.02	0.0738 ± 0.01	0.7762 ± 0.02	0.2832 ± 0.01
KM (all)	0.2652 ± 0.02	0.1494 ± 0.01	0.2719 ± 0.02	0.0833 ± 0.01	0.7424 ± 0.04	0.2933 ± 0.01
Co-train	0.2829 ± 0.03	0.1623 ± 0.02	0.2883 ± 0.03	0.1002 ± 0.02	0.7756 ± 0.03	$\mathbf{0.3327 \pm 0.03}$
Co-reg	0.1622 ± 0.04	0.0695 ± 0.04	0.1666 ± 0.04	0.0200 ± 0.02	0.2425 ± 0.09	0.2276 ± 0.04
RMKMC	0.2873 ± 0.02	0.1755 ± 0.01	0.2973 ± 0.01	0.1057 ± 0.01	0.8364 ± 0.00	0.2989 ± 0.01
DEKM	0.2826 ± 0.01	0.1672 ± 0.00	0.2985 ± 0.01	0.1020 ± 0.01	0.8331 ± 0.00	0.2935 ± 0.01
CaMVC	0.2791 ± 0.02	0.1734 ± 0.01	0.2943 ± 0.02	0.0943 ± 0.01	0.8229 ± 0.01	0.2964 ± 0.01
MVCOR	$\mathbf{0.2935 \pm 0.01}$	$\mathbf{0.1784 \pm 0.01}$	$\mathbf{0.3040 \pm 0.01}$	$\mathbf{0.1070 \pm 0.01}$	$\mathbf{0.8356 \pm 0.00}$	0.3061 ± 0.01

Table 6
The *p*-value of significant difference test between the accuracy of MVCOR and the accuracies of other algorithms in each datasets.

Datasets	Co-reg	Co-train	RMKMC	DEKM	CaMVC
SensIT-3000	0.75	0.43	0.00	0.00	0.00
MSRC-v1	0.00	0.02	0.07	0.00	0.04
AWA-10	0.00	0.10	0.11	0.01	0.00

the other. To generate *attribute-outliers*, we randomly extract an instance and replace its features in two views as random values. In our experiment, 10% instances are preprocessed and labeled as outliers, and two cases when two types of outliers have different outlier ratios are investigated.

6.2.2. Evaluation metrics

For outlier detection, one of the most widely used evaluation approaches is Receiver Operating Characteristic (ROC) analysis, and its evaluation metric is the area under the ROC curve (AUC) [33]. The metric ranges in [0, 1], and higher AUC corresponding to better algorithm performance.

Table 7
Data setting in UCI datasets.

Datasets	Iris	Letter	Ionosphere	Zoo	Waveform	Pima	Wdbc
N	150	1300	351	101	1200	768	569
K	3	26	2	7	3	2	2

6.2.3. Comparison algorithms and experimental setup

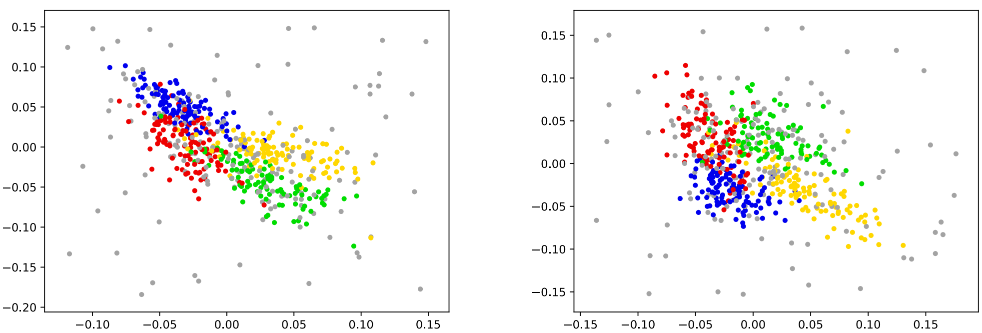
There are two outlier score methods used in this experiment:

- *HOAD* [34] is the horizontal anomaly detection, which is a spectral clustering based method.
- *AP* [35] is the anomaly detection based on affinity propagation, which is the state-of-the-art multi-view outlier detection method. The best anomaly score calculation strategy of AP is *Hilbert–Schmidt Independence Criterion*(HSIC).

The two algorithms above need a similarity matrix, we choose the Gaussian kernel to build it. For each dataset, we repeatedly generate two types of outliers for 30 times and then calculate the mean and standard deviation of AUC value. For our method,

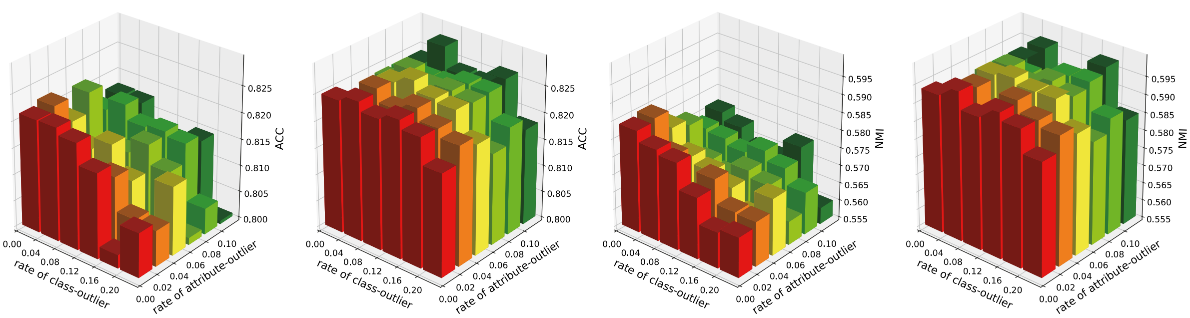
Table 8
The average AUC ($\pm$ STD) in UCI datasets (***C*** represents the *class-outlier* and ***A*** represents the *attrubute-outliers*).

	5% ***C*** + 5% ***A***			3.3% ***C*** + 6.6% ***A***			0% ***C*** + 10% ***A***		
	HOAD	AP	MVCOR	HOAD	AP	MVCOR	HOAD	AP	MVCOR
Iris	0.5941 ± 0.08	$\mathbf{0.9500 \pm 0.03}$	0.8511 ± 0.05	0.5487 ± 0.09	$\mathbf{0.9645 \pm 0.02}$	0.8741 ± 0.05	0.5645 ± 0.08	0.9425 ± 0.03	$\mathbf{0.9629 \pm 0.02}$
Letter	0.6332 ± 0.04	0.8391 ± 0.02	$\mathbf{0.8829 \pm 0.02}$	0.7723 ± 0.04	0.8041 ± 0.01	$\mathbf{0.8655 \pm 0.01}$	0.6263 ± 0.12	0.8279 ± 0.02	$\mathbf{0.9964 \pm 0.00}$
Ionosphere	0.6945 ± 0.09	$\mathbf{0.8935 \pm 0.02}$	0.6530 ± 0.04	0.5289 ± 0.09	$\mathbf{0.8033 \pm 0.02}$	0.5679 ± 0.03	0.1312 ± 0.23	0.8324 ± 0.01	$\mathbf{0.9522 \pm 0.01}$
Zoo	0.4798 ± 0.01	$\mathbf{0.8693 \pm 0.04}$	0.7843 ± 0.06	0.2607 ± 0.09	0.8865 ± 0.03	$\mathbf{0.9310 \pm 0.04}$	0.3005 ± 0.10	0.8602 ± 0.02	$\mathbf{0.9568 \pm 0.05}$
Waveform	0.5976 ± 0.04	$\mathbf{0.8880 \pm 0.05}$	0.8688 ± 0.02	0.7255 ± 0.05	0.8978 ± 0.02	$\mathbf{0.9170 \pm 0.01}$	0.1180 ± 0.06	0.9598 ± 0.00	$\mathbf{0.9899 \pm 0.00}$
Pima	0.6915 ± 0.03	$\mathbf{0.7891 \pm 0.03}$	0.7303 ± 0.02	0.7785 ± 0.03	0.7751 ± 0.02	$\mathbf{0.8602 \pm 0.02}$	0.8411 ± 0.04	0.6982 ± 0.02	$\mathbf{0.9481 \pm 0.01}$
Wdbc	0.7580 ± 0.05	$\mathbf{0.9709 \pm 0.01}$	0.9126 ± 0.03	0.5130 ± 0.04	$\mathbf{0.9723 \pm 0.01}$	0.9178 ± 0.02	0.9363 ± 0.02	0.9298 ± 0.00	$\mathbf{0.9899 \pm 0.00}$



(a) (b)

Fig. 4. Two views of the synthetic dataset. The outliers are denoted as gray points.



(a) ACC (b) NMI

Fig. 5. The clustering performance on the synthetic dataset. The X-coordinate and the Y-coordinate are the rate of the two types of outliers adding in the experiment (e.g. 0.1 is adding 60 outliers). The left picture represents the clustering performance of RMKMC, and the right picture is MVCOR. (a) shows the accuracy result, and (b) shows the NMI result.

γ is set to 2, and we firstly run Algorithm 2 for each outlier generation. And then run Algorithm 1 with relevant θ_1 and θ_2 for 60 times and choose half of the results that objective values are smaller. Finally, we use the number of times that instance is classified into outliers as the outlier score.

6.2.4. Experimental results

The experimental result is reported in Table 8. It is obvious that MVCOR performs better than HOAD in all datasets, and its ability of outlier detection is almost equal to AP. MVCOR has a superior ability to find the *attribute-outlier* and a slightly inferior ability to find the *class-outlier*. The K-means algorithm can only find the globular clusters, which makes MVCOR difficult to extract the complex cluster structure and furthermore accurately identify the *class-outliers*. But the *attribute-outliers* can be easily found even if the method ignores the cluster structure and uses the globular cluster to fit it, so the simple outlier removal strategy is more suitable to find it. MVCOR is not specialized in outlier detection, and it can still correctly find the two types of outliers as well as use the outliers information to improve the clustering result.

6.3. Robustness to outliers

In this subsection, MVCOR is verified that it is more robust than the multi-view clustering method using $L_{2,1}$-norm only.

6.3.1. Datasets

For explanatory purposes, we conduct this experiment on the synthetic dataset. This dataset is generate by the function make_classification in python module *scikit-learn*, contains 600 instances, 4 clusters, 4 features, divided into 2 views, and each view contains 2 features. We use the same method in the last subsection to generate the two types of outliers. Then we get the synthetic dataset with 420 normal instances, 120 *class-outliers* and 60 *attribute-outliers*. The data distribution is shown in Fig. 4.

6.3.2. Experimental setup

We run MVCOR and RMKMC [16] on the dataset. In the beginning, we just use the 420 normal instances. And then, we add *class-outliers* and *attribute-outliers* little by little, as well as observe the clustering performance on the 420 normal instances. MVCOR uses Algorithm 2 to search the appropriate parameters.

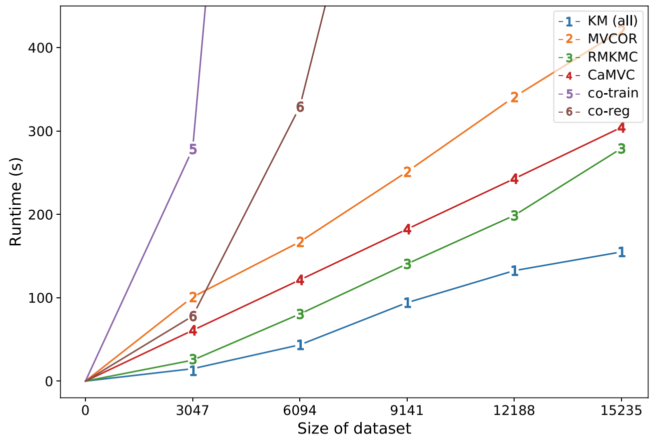


Fig. 6. The runtimes of different multi-view algorithms. DEKM runs 5 times than the proposed method and does not arise in the chart.

6.3.3. Experimental results

The experimental result is shown in Fig. 5. We observe that with the number of both *class-outliers* and *attribute-outliers* increasing, the clustering performance of RMKMC degrades sharply. However, MVCOR avoids this situation by setting correct parameters θ_1 and θ_2. Thus outlier removal strategy can both improve the clustering performance and enhance the robustness of our method. We observe that MVCOR keep a more stable performance as there are fewer outliers in most cases, which empirically demonstrates that MVCOR is more robust than the method using $L_{2,1}$-norm only.

6.4. Runtime analysis

In this subsection, we will show that MVCOR scales almost linearly with the data size. In our experiment, the stop criteria is defined as following (but for co-train and co-reg, we run for the same certain iterations):

$$\frac{|\mathrm{obj}^{t+1} - \mathrm{obj}^{t}|}{|\mathrm{obj}^{t}|} < 10^{-2} \tag{39}$$

where obj^t is the objective value in the tth iteration. $\mathrm{n}_{\max}$ in MVCOR is set to 0.1 number of instances. We randomly sample different numbers of instance from AWA dataset in the list $[3047, 6094, 9141, 12188, 15235]$, and then run all the algorithms used in Section 6.1 while record the runtimes.

The result can be seen in Fig. 6. It is noted that the proposed method MVCOR cost a nearly linear time complexity with the data size, which can be easily applied to large-scale datasets. Meanwhile, it verifies the correctness of theoretical analysis on time complexity, i.e., $\mathrm{O}(\mathrm{tNMKd}_{\max} + \mathrm{tN}\log\mathrm{N})$, and the log term indeed affects slightly. On the other hand, the outlier removal strategy introduces a larger linear coefficient term, so MVCOR works slightly slower than other linear models, while these extra costs contribute to the improvement of the clustering performance. For the graph-based method, co-train and co-reg work well on small datasets, but when the size of dataset is scaled up, the time cost increases rapidly, as well as the algorithms no longer has any effect in clustering soon. Besides, the construction of kernel or similarity matrices costs the $\mathrm{O}(\mathrm{N}^2)$ space complexity, which is also unacceptable in large-scale dataset.

7. Conclusion

In this paper, we propose a robust multi-view k-means method with outlier detection to remove the *class-outlier* and *attribute-outliers* simultaneously. To remove the negative influence of outliers in multi-view data, we introduce two clusters to hold these two types of outliers. Besides, our method inherits the effectiveness of the classical k-means algorithm, with a low time complexity which is the main advantage to analyze real-world datasets. We also introduce an efficient alternating minimization algorithm to optimize the model, and strictly prove its convergence. Experimental results show that the proposed method MVCOR performs well in clustering when the outliers appear a percentage of the datasets, and can easily deal with the situation that part of the outliers is *class-outliers* but other robust clustering methods cannot do it. Also, MVCOR performs nearly equal to the state-of-the-art method AP in outlier detection, illustrating that the proposed method can find outliers correctly. However, the algorithm based on k-means can only deal with the spherical clusters. MVCOR is also based on k-means, thus it is not suitable for the dataset with irregular shapes, which is left for future research.

CRediT authorship contribution statement

Chuan Chen: Conceptualization, Methodology, Resources. **Yu Wang:** Data curation, Writing - original draft, Software. **Weibo Hu:** Validation, Visualization, Investigation. **Zibin Zheng:** Supervision, Writing - review & editing.

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.

Acknowledgment

This work was supported by the National Key Research and Development Program (2016YFB1000101), the National Natural Science Foundation of China (11801595), Key-Area Research and Development Program of Guangdong Province (2020B010165003), the Guangdong Basic and Applied Basic Research Foundation (2019A1515011043), and the Natural Science Foundation of Guangdong (2018A030310076).

References

[1] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: Recent progress and new challenges, Inf. Fusion 38 (2017) 43–54.
[2] F. Nie, J. Li, X. Li, et al., Self-weighted Multiview Clustering with Multiple Graphs, in: IJCAI, 2017, pp. 2564–2570.
[3] C.-D. Wang, J.-H. Lai, S. Philip, Multi-view clustering based on belief propagation, IEEE Trans. Knowl. Data Eng. 28 (4) (2015) 1007–1021.
[4] F. Ye, Z. Chen, H. Qian, R. Li, C. Chen, Z. Zheng, New approaches in multi-view clustering, in: Recent Applications in Data Clustering, BoD–Books on Demand, 2018, p. 195.
[5] C. Chen, H. Qian, W. Chen, Z. Zheng, H. Zhu, Auto-weighted multi-view constrained spectral clustering, Neurocomputing (2019).
[6] C. Chen, J. Xin, Y. Wang, L. Chen, M.K. Ng, A semisupervised classification approach for multidomain networks with domain selection, IEEE Trans. Neural Netw. Learn. Syst. 30 (1) (2018) 269–283.
[7] H. Wang, Y. Yang, B. Liu, H. Fujita, A study of graph-based system for multi-view clustering, Knowl.-Based Syst. 163 (2019) 1009–1019.
[8] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 977–986.
[9] Z. Zhang, L. Liu, F. Shen, H.T. Shen, L. Shao, Binary multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. (2018).
[10] Z. Chen, C. Chen, Z. Zheng, Y. Zhu, Tensor Decomposition for Multilayer Networks Clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3371–3378.
[11] Z. Kang, G. Shi, S. Huang, W. Chen, X. Pu, J.T. Zhou, Z. Xu, Multi-graph fusion for multi-view spectral clustering, Knowl.-Based Syst. 189 (2020) 105102.

[12] S. Huang, Y. Ren, Z. Xu, Robust multi-view data clustering with multi-view capped-norm k-means, Neurocomputing 311 (2018) 197–208.
[13] X. Tao, R. Wang, R. Chang, C. Li, R. Liu, J. Zou, Spectral clustering algorithm using density-sensitive distance measure with global and local consistencies, Knowl.-Based Syst. 170 (2019) 26–42.
[14] X. Liu, Y. Dou, J. Yin, L. Wang, E. Zhu, Multiple Kernel k-Means Clustering with Matrix-Induced Regularization, in: AAAI, 2016, pp. 1888–1894.
[15] J. Wu, H. Liu, H. Xiong, J. Cao, J. Chen, K-means-based consensus clustering: A unified view, IEEE Trans. Knowl. Data Eng. 27 (1) (2014) 155–169.
[16] X. Cai, F. Nie, H. Huang, Multi-view K-means clustering on big data, in: International Joint Conference on Artificial Intelligence, 2013, pp. 2598–2604.
[17] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.
[18] X. Cui, P. Zhu, X. Yang, K. Li, C. Ji, Optimized big data K-means clustering using mapreduce, J. Supercomput. 70 (3) (2014) 1249–1259.
[19] S. Bickel, T. Scheffer, Multi-view clustering, in: ICDM, Vol. 4, 2004, pp. 19–26.
[20] X. Chen, X. Xu, J.Z. Huang, Y. Ye, TW-K-means: Automated two-level variable weighting clustering algorithm for multiview data, IEEE Trans. Knowl. Data Eng. 25 (4) (2011) 932–944.
[21] X. Zhao, N. Evans, J.-L. Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognit. Lett. 41 (2014) 73–82.
[22] Y. Ye, X. Liu, J. Yin, E. Zhu, Co-regularized kernel k-means for multi-view clustering, in: Pattern Recognition (ICPR), 2016 23rd International Conference on, IEEE, 2016, pp. 1583–1588.
[23] G.-Y. Zhang, C.-D. Wang, D. Huang, W.-S. Zheng, Y.-R. Zhou, TW-Co-k-means: Two-level weighted collaborative k-means for multi-view clustering, Knowl.-Based Syst. 150 (2018) 127–138.
[24] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, T.-J. Chen, Fuzzy c-means clustering with spatial information for image segmentation, Comput. Med. Imaging Graph. 30 (1) (2006) 9–15.
[25] N. Dhanachandra, K. Manglem, Y.J. Chanu, Image segmentation using K-means clustering algorithm and subtractive clustering algorithm, Procedia Comput. Sci. 54 (2015) 764–771.
[26] M. Mahdavi, H. Abolhassani, Harmony K-means algorithm for document clustering, Data Min. Knowl. Discov. 18 (3) (2009) 370–391.
[27] P. Jana, S. Ghosh, S.K. Bera, R. Sarkar, Handwritten document image binarization: An adaptive K-means based approach, in: 2017 IEEE Calcutta Conference, CALCON, IEEE, 2017, pp. 226–230.
[28] M.E. Celebi, Improving the performance of k-means for color quantization, Image Vis. Comput. 29 (4) (2011) 260–271.
[29] J. Deng, J. Guo, Y. Wang, A novel K-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering, Knowl.-Based Syst. 175 (2019) 96–106.
[30] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint l2, 1-norms minimization, in: Advances in Neural Information Processing Systems, 2010, pp. 1813–1821.
[31] G. Gan, M.K.-P. Ng, K-means clustering with outlier removal, Pattern Recognit. Lett. 90 (2017) 8–14.
[32] D.M. Hawkins, Identification of Outliers, Vol. 11, Springer, 1980.
[33] S. Li, M. Shao, Y. Fu, Multi-view low-rank analysis with applications to outlier detection, ACM Trans. Knowl. Discov. Data 12 (3) (2018) 1–22.
[34] J. Gao, W. Fan, D. Turaga, S. Parthasarathy, J. Han, A spectral framework for detecting inconsistency across multi-source object relationships, in: 2011 IEEE 11th International Conference on Data Mining, IEEE, 2011, pp. 1050–1055.
[35] A.M. Alvarez, M. Yamada, A. Kimura, T. Iwata, Clustering-based anomaly detection in multi-view data, in: ACM International Conference on Conference on Information & Knowledge Management, 2013, pp. 1545–1548.
[36] H. Zhao, Y. Fu, Dual-regularized multi-view outlier detection, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
[37] H. Zhao, H. Liu, Z. Ding, Y. Fu, Consensus regularized multi-view outlier detection, IEEE Trans. Image Process. 27 (1) (2018) 236–248.
[38] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the 28th International Conference on Machine Learning, ICML-11, 2011, pp. 393–400.
[39] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Advances in Neural Information Processing Systems, 2011, pp. 1413–1421.
[40] F. Nie, G. Cai, J. Li, X. Li, Auto-weighted multi-view learning for image clustering and semi-supervised classification, IEEE Trans. Image Process. 27 (3) (2017) 1501–1511.
[41] H. Wang, F. Nie, H. Huang, Multi-view clustering and feature learning via structured sparsity, in: International Conference on Machine Learning, 2013, pp. 352–360.
[42] D. Kong, C. Ding, H. Huang, Robust nonnegative matrix factorization using l21-norm, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, ACM, 2011, pp. 673–682.
[43] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, Y.-D. Shen, Robust Multiple Kernel K-means Using L21-Norm, in: IJCAI, 2015, pp. 3476–3482.
[44] R. Kannan, H. Woo, C.C. Aggarwal, H. Park, Outlier detection for text data, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 489–497.
[45] B. Jiang, C. Ding, Outlier regularization for vector data and L21 norm robustness, 2017, arXiv preprint arXiv:1706.06409.
[46] H. Gao, F. Nie, W. Cai, H. Huang, Robust capped norm nonnegative matrix factorization: Capped norm nmf, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 871–880.
[47] 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.
[48] B. Liu, Y. Xiao, S.Y. Philip, Z. Hao, L. Cao, An efficient approach for outlier detection with imperfect data labels, IEEE Trans. Knowl. Data Eng. 26 (7) (2013) 1602–1616.
[49] E.M. Knox, R.T. Ng, Algorithms for mining distancebased outliers in large datasets, in: Proceedings of the International Conference on Very Large Data Bases, Citeseer, 1998, pp. 392–403.
[50] E.M. Knorr, R.T. Ng, V. Tucakov, Distance-based outliers: algorithms and applications, VLDB J. Int. J. Very Large Data Bases 8 (3–4) (2000) 237–253.
[51] F. Angiulli, S. Basta, C. Pizzuti, Distance-based detection and prediction of outliers, IEEE Trans. Knowl. Data Eng. 18 (2) (2005) 145–160.
[52] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: ACM Sigmod Record, Vol. 29, No. 2, ACM, 2000, pp. 93–104.
[53] V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, P. Fränti, Improving K-means by outlier removal, in: H. Kalviainen, J. Parkkinen, A. Kaarna (Eds.), Image Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 978–987.
[54] 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.
[55] H. Liu, J. Li, Y. Wu, Y. Fu, Clustering with outlier removal, 2018, arXiv preprint arXiv:1801.01899.
[56] V.P. Janeja, R. Palanisamy, Multi-domain anomaly detection in spatial datasets, Knowl. Inf. Syst. 36 (3) (2013) 749–788, http://dx.doi.org/10.1007/s10115-012-0534-5.
[57] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.
[58] J. Winn, N. Jojic, Locus: Learning object classes with unsupervised segmentation, in: Tenth IEEE International Conference on Computer Vision, Vol. 1, ICCV'05, IEEE, 2005, pp. 756–763.
[59] C.H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 951–958.
[60] T. Li, C. Ding, The relationships among various nonnegative matrix factorization methods for clustering, in: Sixth International Conference on Data Mining, ICDM'06, IEEE, 2006, pp. 362–371.
[61] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (Dec) (2002) 583–617.
[62] J. Xu, J. Han, F. Nie, Discriminatively embedded k-means for multi-view clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5356–5364.