Contents lists available at ScienceDirect

Information Fusion

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

Multi-view Outlier Detection via Graphs Denoising

Boao Hu [a], Xu Wang [a], Peng Zhou [a,*], Liang Du [b]

[a] Anhui Provincial International Joint Research Center for Advanced Technology in Medical Imaging, School of Computer Science and Technology, Anhui University, Hefei 230601, China
[b] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

ARTICLE INFO

Keywords:
Multi-view learning
Outlier detection
Multiple graph learning

ABSTRACT

Recently, multi-view outlier detection attracts increasingly more attention. Although existing multi-view outlier detection methods have demonstrated promising performance, they still suffer from some problems. Firstly, many methods make the assumption that the data have a clear clustering structure and detect the outliers by using some off-the-shelf clustering methods. Therefore, the performance of these methods depends on the clustering methods they used, and thus these methods are hard to handle complicated data. Secondly, some methods ignore the complicated structure or distribution of class outliers and directly learn a consensus representation by simply combining the representation of different views linearly. To tackle these problems, we propose a novel method named Multi-view Outlier Detection with Graph Denoising (MODGD). We first construct a graph for each view, and then learn a consensus graph by ensembling the multiple graphs. When fusing the multiple graphs, we explicitly characterize and extract the structured outliers on each graph and recover the multiple clean graphs for the ensemble. During the process of multiple graph denoising and fusion, we carefully design an outlier measurement criterion based on the characteristics of attribute and class outliers. The extensive experiments on benchmark data sets demonstrate the effectiveness and superiority of the proposed method. The codes of this paper are released in http://Doctor-Nobody.github.io/codes/MODGD.zip.

1. Introduction

Unsupervised outlier detection is a challenging task in data mining and machine learning. It identifies the outliers in collected data without labels, which has attracted a lot of attention [1–8]. Unsupervised outlier detection has been widely applied in many fields, such as urban traffic [9], social media [10,11], and fraud detection [12]. In past decades, a massive number of outlier detection methods have been proposed, e.g., graph-based methods, clustering-based methods, distance-based methods, and density-based methods [13]. In recent years, deep learning has made significant progress in many tasks of machine learning due to its ability to learn the representation of complex data. A lot of deep unsupervised outlier detection models have been proposed [14–20], and demonstrate promising performance.

Although these different types of outlier detection methods have demonstrated promising performance, they only focus on data from one source, i.e., single-view data. However, nowadays obtaining data from different sources is no longer as difficult as before. These data are called multi-view data, where features from a specific source are considered as a specific view. These views characterize the object from different perspectives. Therefore, compared to single-view data, these multi-view data contain more rich information. How to detect outliers on such multi-view data attracts increasingly more attention.

To tackle this problem, multi-view outlier detection methods have been proposed. In the single-view outlier detection task, one basic assumption is that normal instances should be close to the majority of instances [21,22]. However, in multi-view data, even though one instance is close to the majority of instances, if it behaves inconsistently among all views, it may still be an outlier. Therefore, the definition of the outlier in multi-view data is different from that in single-view data. Three widely used types of outliers on multi-view data in previous literature are defined as follows [23,24]:

- **Class outlier**: The behaviors of this type of outlier are inconsistent among all views. Specifically, the outliers in this type are close to their neighbors, but they may belong to different clusters or classes in different views.
- **Attribute outlier**: The characteristic of this type of outlier is that it exhibits great difference to most other instances in each view, namely consistent abnormal behaviors.

* Corresponding author.
E-mail addresses: e21201093@stu.ahu.edu.cn (B. Hu), e22301279@stu.ahu.edu.cn (X. Wang), zhoupeng@ahu.edu.cn (P. Zhou), duliang@sxu.edu.com (L. Du).

https://doi.org/10.1016/j.inffus.2023.102012
Received 5 June 2023; Received in revised form 17 August 2023; Accepted 5 September 2023
Available online 9 September 2023

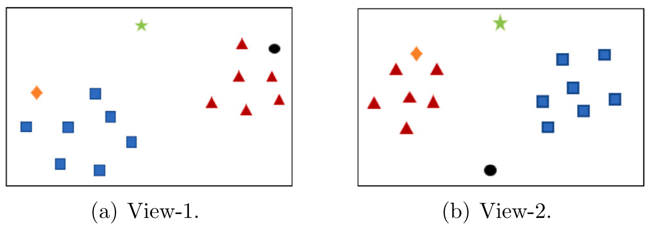


Fig. 1. Illustration of three types of outliers: (1) Class outlier (the orange diamond) belongs to the cluster of the blue squares in the first view, while it is similar to the red triangles in the second view; (2) Attribute outlier (the green star) is dissimilar to the majority of instances in both views; (3) Class-Attribute outlier (the black circle) is similar to red triangles in first view and is remote from the other instances in another view.

- **Class-Attribute outlier**: This type of outlier is the mixture of the above two types. It means that, in some views, they have the characteristics of the class outliers, while in other views they behave like the attribute outliers.

Fig. 1 shows an example of the three types of outliers in multi-view data. The blue squares and red triangles denote two clusters in two views. The orange diamond is one of the class outlier, because it belongs to the cluster of blue squares in the first view but belongs to the cluster of red triangles in the second view. The green star is one of the attribute outliers because it is far from other instances in both two views. The black dot is one of the class-attribute outliers because it is a class outlier in the first view and is an attribute outlier in the second view.

To detect these types of outliers on multi-view data, some multi-view outlier detection methods have been proposed [23–31]. Although these methods have achieved a certain extent of success, they still have some shortcomings. First, most methods are based on the cluster structure of data [24–29]. The performance of these methods depends on the performance of the used clustering methods and may fail when the data do not have a clear cluster structure or when the used clustering method cannot reveal the cluster structure of data. Secondly, some methods, such as [24,28,31], detect the outliers by learning a consensus representation from all views. However, when learning the consensus representation, they simply linearly combine the information in multiple views without paying more attention to the complicated structure or distribution of the outliers. Thirdly, some methods first learn the representations of data in a specific view, and then use pair-wise constraint between two views to align all representations of data [25–27,29]. Such approaches are often prone to pay more attention to one specific view and thus cannot fully utilize the abundant information of multi-view data. If the quality of the specific view is not good, the performance of multi-view learning may also be limited. Besides, some methods cannot handle all three types of outliers, such as [26,27].

To overcome these shortcomings of existing methods, we propose a method named Multi-view Outlier Detection with Graphs Denoising (MODGD). To avoid the clustering assumption, we apply the graph to reveal the structure of the data. In more detail, we first construct a graph for each view. Since the attribute outliers are far away from other data, we detect the attribute outliers on the graph by observing the similarity between them and their neighbors. In this way, attribute outliers can be detected even if there is no clear clustering structure in the data. To identify the class outliers, we discover the inconsistent behavior by learning a consensus graph from the multiple graphs. Different from conventional multiple graph learning, which directly ensembles multiple graphs, we ensemble them by recovering multiple clean graphs via graph denoising. We observe that the class outliers on the graph have some special structure that can help us to detect them. Specifically, if an instance is an outlier, both the corresponding row and column in the adjacency matrix of the graph are simultaneously contaminated, which means the outliers on the graph are symmetric and row-wise and column-wise. Then, we can extract such row-wise and column-wise outliers and recover multiple clean graphs for fusion. During the process of multiple graph denoising and fusion, we can directly identify the class outliers. At last, we design an outlier measurement criterion to score each instance according to its probability of being an outlier.

The main contributions of this paper are summarized as follows:

- We propose a novel unsupervised multi-view outlier detection method, which can detect all three types of outliers.
- Different from existing multi-view learning methods, which simply ensembling the information of different views linearly, we propose a new multiple graph learning method that can directly characterize the structured outliers (i.e., the symmetric and row-wise and column-wise outliers) on the graph.
- The extensive experiments on benchmark multi-view data show that the proposed method outperforms the state-of-the-art multi-view outlier detection methods.

2. Related work

Unsupervised outlier detection is a fundamental problem in data mining and machine learning. It aims to accurately identify the outliers in the data set. Nevertheless, most existing works only focus on single-view data. In the single-view setting, the definition of outliers is simple. The outliers are the instances that are far away from the majority instances, which is similar to the attribute outliers in the multi-view setting. In the single-view setting, there are no class outliers. However, in the multi-view setting, the cases are more complicated. In addition to the attribute outliers, which are similar to the outliers in the single-view setting, the multi-view setting also has class outliers, which are the instances having different behaviors in each view. Combining the attribute outliers and the class outliers, there exists the third type of outliers, i.e., class-attribute outliers. Therefore, the outliers in the multi-view setting are much more complicated than those in the single-view setting.

In the past decade, several multi-view methods have been proposed. For example, Gao et al. [26] first extended outlier detection from single-view to multiple views and proposed a method named HOAD. It applied spectral clustering in each view, while instances were constrained to be allocated to the same cluster across multiple views. The inconsistency of instances was used to determine whether ones were outliers. Marcos Alvarez et al. [27] proposed a method named APOD, it performed affinity propagation clustering separately in the different views and obtained the affinity vectors for each object. Then, it identified the outliers by comparing instances' affinity vectors in the multiple views. The frameworks of HOAD and APOD are both simple and only focus on class outliers while ignoring other types of outliers.

Furthermore, Zhao et al. [25] expanded the definition of the outlier in multiple views and proposed the method DMOD. It first studied the attribute outlier in multi-view outlier detection. Based on the framework of k-means, it represented the multi-view data with latent coefficients and construction errors. A pair-wise constraint between two views was used to align all representations. Moreover, a well-designed outlier measurement criterion was proposed in this paper. Another method MLRA [29] obtained the coefficient matrix of each view, and then minimized their rank. Similarly, coefficient matrices were also aligned by pair-wise constraint. These methods, HOAD, APOD, DMOD, and MLRA are all based on pair-wise constraints which cannot fully utilize the information of multiple view data.

Li et al. [24] supplemented the definition of a class-attribute outlier in multi-view outlier detection and proposed the LDSR method. To overcome the shortcomings of pair-wise constraint, it separated the representations as consensus and residual parts. The relationship between CRMOD [28] and DMOD is similar to the one between MLRA and

Table 1
Notations used in our method and their descriptions.

Notations	Descriptions
n	Number of instances.
V	Number of views.
$\mathbf{X}^v \in \mathbb{R}^{d^v \times n}$	The feature matrix of the vth view.
$\mathbf{x}_i^v \in \mathbb{R}^{d^v}$	The ith instance in the vth view.
$\mathbf{S}^v \in \mathbb{R}^{n\times n}$	The adjacency matrix of the vth graph.
$\hat{\mathbf{S}}^v \in \mathbb{R}^{n\times n}$	Normalized adjacency matrix of the vth graph.
$\mathbf{S} \in \mathbb{R}^{n\times n}$	The consensus matrix.
$\mathbf{W} \in \mathbb{R}^{n\times n}$	The weight matrix.
$\mathbf{L} \in \mathbb{R}^{n\times n}$	The Laplacian matrix of $\mathbf{S}$.
$\mathbf{E}^v \in \mathbb{R}^{n\times n}$	The row-wise sparse outlier matrix.
$\mathbf{s}^n \in \mathbb{R}^n$	The neighborhood score.
$\mathbf{s}^c \in \mathbb{R}^n$	The consistency score.
$\mathbf{s} \in \mathbb{R}^n$	The final score.

LDSR. CRMOD minimized the disagreement among the consensus representation and others. Nevertheless, these methods mentioned above all depend on some standard clustering methods, which means they may fail if there is no clear cluster structure in the data or if the used clustering method is inappropriate for the given data set.

Recently, some methods based on the local similarity between the instance and its neighbors have been proposed. In this way, these methods can handle data without clustering methods. In MUVAD [30], Sheng et al. proposed a nearest neighbor-based outlier measurement criterion. Then, they designed an objective function according to this criterion to find normal instances. Based on an autoencoder network, Cheng et al. [23] proposed a neighborhood consensus network in NCMOD. Through the autoencoder, it mapped instances to a latent space for each view. Wang et al. designed SRLSP [31] method which contained three terms. First, with the adaptive similarity learning term, it obtained the similarity between an instance and its neighbors in each view. Then, it acquired the consensus similarity by a graph fusion term. Finally, it minimized the reconstruction error by a self-representation term. However, similar to other methods of learning consistent representations (i.e., LDSR and CRMOD), it failed to characterize the structure of class outliers, and obtained the consensus result by a simple linear combination of the information in multiple views.

3. Multi-view outlier detection with graphs denoising

In this section, we introduce our method in detail. First, we introduce some notations. We denote a multi-view data set $\mathcal{X}$ with n instances and V views as $\mathcal{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^V\}$, where $\mathbf{X}^v = \{\mathbf{x}_1^v, \mathbf{x}_2^v, \dots, \mathbf{x}_n^v\} \in \mathbb{R}^{d^v\times n}$ is the feature matrix of the vth view and each column of this matrix represents one instance. d^v refers to the feature dimension of the vth view. In this paper, $\mathbf{A}_{i.}$ and $\mathbf{A}_{.i}$ denote the ith row and column vector of matrix $\mathbf{A}$, respectively. $\mathbf{A}_{ij}$ is the (i,j)-th element of $\mathbf{A}$. The main notations used in our method and their descriptions are summarized in Table 1.

The task of unsupervised outlier detection is to score each instance according to its probability of being an outlier. A common approach is to design two scores to detect attribute outliers and class outliers, respectively, and then obtain the final score by combining these two scores. In our method, we calculate two scores named *neighborhood score* $\mathbf{s}^n \in \mathbb{R}^n$ to detect the attribute outliers, and *consistency score* $\mathbf{s}^c \in \mathbb{R}^n$ to detect the class outliers. We first construct a k-nearest neighbors (KNN) graph for each view, and calculate the instances' probability of being the attribute outlier from the instance and its top-k neighbors as the neighborhood score. Then, we normalize these graphs and calculate the consistency score by carefully characterizing the structure of the class outliers. Fig. 2 shows the entire process of the method. In the following, we will introduce our method in more detail.

3.1. Calculating the neighborhood score

We calculate the neighborhood score $\mathbf{s}^n$ to detect the attribute outliers. When detecting the attribute outliers, we follow the assumption used in single-view outlier detection that is the outliers are often far away from other instances. It means that, compared with normal instances, the outliers are far away from their neighbors. To this end, we use neighborhood information to detect the attribute outliers.

In more detail, given a data set with V views, we first construct the KNN graph for each view. Specifically, in the vth view, we find each instance's top-k similar neighbors and set an edge between it and each selected neighbor. There are many approaches to construct the KNN graph. In our method, we adopt the Heat Kernel to calculate the similarity between the instances $\mathbf{x}_i^v$ and $\mathbf{x}_j^v$. Formally, we construct the KNN graph $\mathcal{G}^v$ whose adjacency matrix $\mathbf{S}^v$ is:

$$S_{ij}^v = \begin{cases} e^{-\frac{\left\|\mathbf{x}_i^v - \mathbf{x}_j^v\right\|_2^2}{2t^2}}, & \text{if} \quad \mathbf{x}_j^v \in \mathcal{N}_i^v, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where t is the bandwidth parameter in heat conduction equation and $\mathcal{N}_i^v$ denotes the neighbor set of $\mathbf{x}_i$. We construct KNN graphs $\{\mathcal{G}^v\}_{v=1}^V$ whose adjacency matrices are $\{\mathbf{S}^v\}_{v=1}^V$ as described above. Then, we can obtain the neighborhood score of an instance with the mean similarity of the instance and its neighbors. More formally, the neighborhood score of $\mathbf{x}_i$, which is denoted as s_i^n, is defined as:

$$s_i^n = \sum_{v=1}^{V} \frac{1}{|\mathcal{N}_i^v|} \sum_{\mathbf{x}_j^v \in \mathcal{N}_i^v} S_{ij}^v \tag{2}$$

The smaller s_i^n is, which means $\mathbf{x}_i$ is less similar to its neighbors, the more likely $\mathbf{x}_i$ is an attribute outlier.

3.2. Calculating the consistency score

When detecting the class outliers, we should discover the inconsistent behaviors of instances. As claimed before, our method does not depend on the standard clustering method and discovers inconsistent behaviors from the intrinsic structure of data. To achieve this, we also use the KNN graphs constructed before and learn a consensus graph $\mathcal{G}$ from these KNN graphs. Then we explore the inconsistent behaviors according to the consensus graph.

One natural way to learn the consensus graph $\mathcal{G}$ is to directly ensemble $\{\mathbf{S}^v\}_{v=1}^V$ to obtain the consensus adjacency matrix. Here, for more effectively ensembling the multiple KNN graphs, we first normalize them. More formally, we normalize $\{\mathbf{S}^v\}_{v=1}^V$ by:

$$\hat{\mathbf{S}}^v = (\mathbf{D}^v)^{-\frac{1}{2}} \mathbf{S}^v (\mathbf{D}^v)^{-\frac{1}{2}} \tag{3}$$

where $\mathbf{D}^v$ is a diagonal matrix with diagonal elements $D_{ii}^v = \sum_j^n S_{ij}^v$. By this normalization, it can maintain the original distribution to prevent significant differences in distribution between high-degree and low-degree vertices in graphs.

Then, we consider how to ensemble $\{\hat{\mathbf{S}}^v\}_{v=1}^V$ to obtain the consensus matrix $\mathbf{S}$. Notice that since the data contains outliers, $\{\hat{\mathbf{S}}^v\}_{v=1}^V$ are also contaminated by the outliers. To effectively obtain a clean consensus matrix, we should recover the clean adjacency matrix for each view. Therefore, we need to take a closer look at the structures of outliers in $\{\hat{\mathbf{S}}^v\}_{v=1}^V$. Assuming that in the vth view $\mathbf{x}_i$ is an outlier, then the ith row of $\hat{\mathbf{S}}^v$ is corrupted and incorrect. Since $\hat{\mathbf{S}}^v$ is symmetric, the ith column of $\hat{\mathbf{S}}^v$ is also corrupted. Therefore, the structure of outliers in the adjacency matrix should be row-wise and column-wise, and symmetric. To this end, we introduce a row-wise sparse outlier matrix $\mathbf{E}^v \in \mathbb{R}^{n\times n}$ to characterize the outliers. Then $\mathbf{E}^{vT}$ is a column-wise sparse matrix. We can use $\mathbf{E}^v + \mathbf{E}^{vT}$ to denote such row-wise and column-wise symmetric outlier matrix for the vth view.

After obtaining the structured outlier matrix, we can recover the clean adjacency matrix for the vth view as $\hat{\mathbf{S}}^v - (\mathbf{E}^v + \mathbf{E}^{vT})$. Then, we

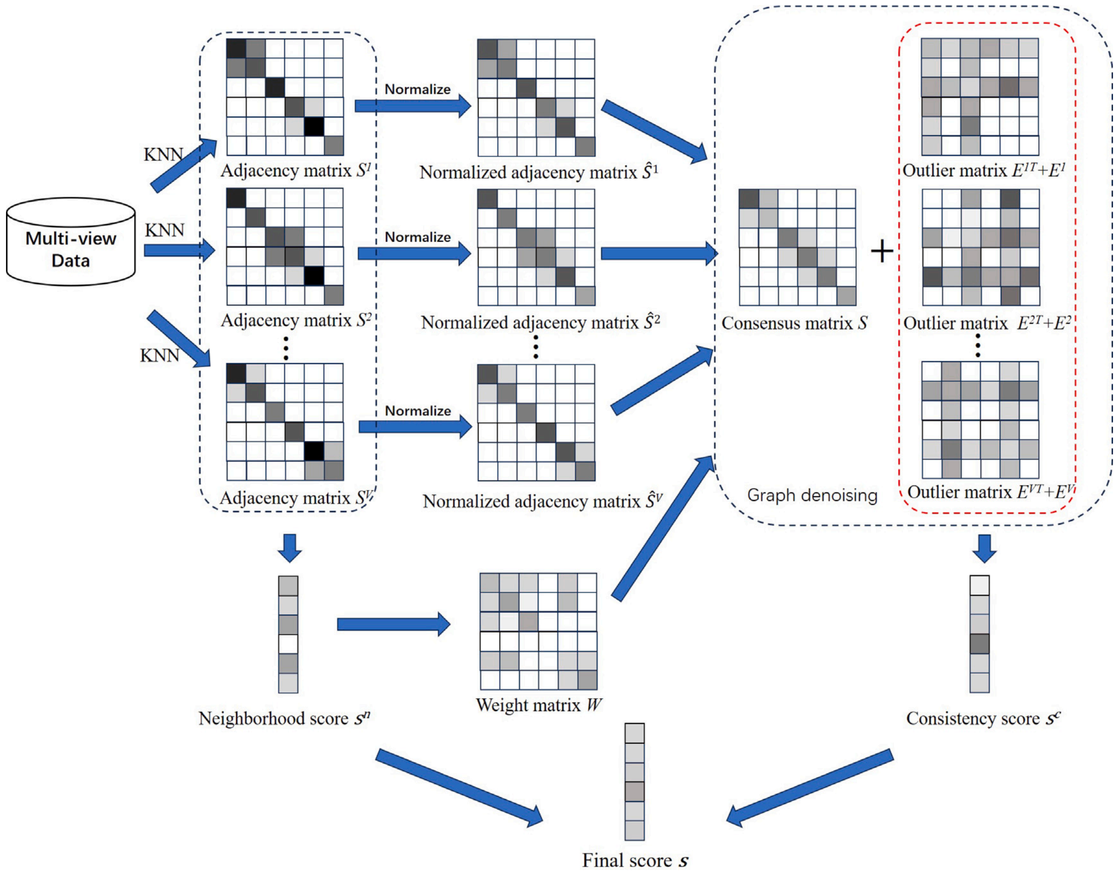


Fig. 2. The flowchart of the proposed MODGD. We first construct V KNN graphs from the multi-view data, and then compute the neighborhood score $\mathbf{s}^n$ from these adjacency matrices $\mathbf{S}^1, \dots, \mathbf{S}^V$. Then we learn the consensus graph $\mathbf{S}$ and the structured outlier matrices $\mathbf{E}^v + \mathbf{E}^{vT}$ from the normalized adjacency matrices. Next, we compute the consistency scores $\mathbf{s}^c$ from these outlier matrices. At last, we combine $\mathbf{s}^n$ and $\mathbf{s}^c$ to get the final outlier score $\mathbf{s}$.

ensemble these $\hat{\mathbf{S}}^v - (\mathbf{E}^v + \mathbf{E}^{vT})$ to obtain the consensus matrix $\mathbf{S}$ as many multi-view learning methods did [32–37]:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{E}^v,\alpha_v} \sum_{v=1}^{V} \alpha_v^2 \left(\left\| \mathbf{S} - \left(\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \right) \right\|_F^2 + \lambda \|\mathbf{E}^v\|_{2,1} \right) \\ &s.t. \quad \mathbf{S} = \mathbf{S}^T, \quad \mathbf{S} \geq \mathbf{0}, \quad \hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \geq \mathbf{0} \\ &\qquad \sum_{v=1}^{V} \alpha_v = 1, \quad \alpha_v \geq 0, \end{aligned} \tag{4}$$

where λ is a balancing parameter. $\ell_{2,1}$-norm is used to ensure the row sparsity of $\mathbf{E}^v$ [38]. The constraints on $\mathbf{S}$ ensure that $\mathbf{S}$ is symmetric and non-negative. The constraint $\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \geq 0$ makes sure that the cleaned matrix $\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT}$ is non-negative and thus is a valid adjacency matrix. α_v is the non-negative weight of the vth view. The constraint $\sum_{i=1}^{V} \alpha_v = 1$ is to avoid trivial solution. The larger α_v is, the higher quality of the vth view is.

Notice that in this step we wish to detect the class outliers by computing the consistency score. However, if we directly optimize Eq. (4), it also involves the attribute outliers because the attribute outliers are still in the adjacency matrices $\{\hat{\mathbf{S}}^v\}_{v=1}^V$. To address this issue, we reformulate Eq. (4) to alleviate the effects of attribute outliers. To this end, we can introduce a weight matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ on $\mathbf{S}$ to make sure that the weights of attribute outliers should be small, which means the attribute outliers will hardly influence the optimization of Eq. (4). Since we already obtain the neighborhood score $\mathbf{s}^n$, we can design the weight W_{ij} for the instance pair $(\mathbf{x}_i, \mathbf{x}_j)$ as the geometric average of the neighborhood score of $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$W_{ij} = \sqrt{s_i^n \cdot s_j^n} \tag{5}$$

Here we use the geometric average because as long as one of $\mathbf{x}_i$ and $\mathbf{x}_j$ is the attribute outliers, W_{ij} will be small and its influence will be alleviated. Then, we impose $\mathbf{W}$ on the ensemble term, leading to $\left\| \mathbf{W} \odot \left(\mathbf{S} - \left(\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \right) \right) \right\|_F^2$, where $\odot$ is the element-wise production.

In addition, to make the consensus matrix have a clearer structure, we impose a low-rank constraint on it. In more detail, we wish the consensus graph $\mathcal{G}$ have a small number of (i.e., c) connective components. Following [39], we have that the number of connective components in $\mathcal{G}$ is equal to the multiplicity of the eigenvalue 0 of the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is a diagonal matrix whose ith diagonal element is $D_{ii} = \sum_{j=1}^{n} S_{ij}$. To this end, we add the constraint $rank(\mathbf{L}) = n - c$ and obtain the final objective function:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{E}^v,\alpha_v} \sum_{v=1}^{V} \alpha_v^2 \left(\left\| \mathbf{W} \odot \left(\mathbf{S} - \left(\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \right) \right) \right\|_F^2 + \lambda \|\mathbf{E}^v\|_{2,1} \right) \\ &s.t. \quad \mathbf{S} = \mathbf{S}^T, \quad \mathbf{S} \geq 0, \quad \hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \geq 0, \\ &\qquad \sum_{v=1}^{V} \alpha_v = 1, \quad \alpha_v \geq 0, \quad rank(\mathbf{L}) = n - c. \end{aligned} \tag{6}$$

Eq. (6) contains the constraint $rank(\mathbf{L}) = n - c$, making it hard to optimize. Fortunately, according to Ky Fan Theorem [40], we can introduce an auxiliary orthogonal matrix $\mathbf{Y} \in \mathbb{R}^{n\times c}$ and a large enough parameter ρ to reformulate it as:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{E}^v,\mathbf{Y},\alpha_v} \sum_{v=1}^{V} \alpha_v^2 \left(\left\| \mathbf{W} \odot \left(\mathbf{S} - \left(\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \right) \right) \right\|_F^2 + \lambda \|\mathbf{E}^v\|_{2,1} \right) + 2\rho tr(\mathbf{Y}^T \mathbf{L}\mathbf{Y}) \\ &s.t. \quad \mathbf{S} = \mathbf{S}^T, \quad \mathbf{S} \ge 0, \quad \hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \ge 0, \quad \mathbf{Y}^T\mathbf{Y} = \mathbf{I}, \\ &\qquad \sum_{v=1}^{V} \alpha_v = 1, \quad \alpha_v \ge 0. \end{aligned} \tag{7}$$

Notice that, ρ is a large enough parameter to make sure that the rank of $\mathbf{L}$ is $n - c$.

Then, we adopt an alternating optimization strategy to optimize each variable in Eq. (7). More specifically, we optimize $\mathbf{S}$, $\mathbf{E}^v$, α_v, $\mathbf{Y}$ respectively by fixing the other variables.

3.2.1. Optimizing S

When optimizing $\mathbf{S}$, we first rewrite Eq. (7) as

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{v=1}^{V} \left(\alpha_v^2 \left\| \mathbf{W} \odot \left(\mathbf{S} - \left(\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \right) \right) \right\|_F^2 \right) + \rho \sum_{p,q=1}^{n} \left\| \mathbf{y}_p - \mathbf{y}_q \right\|_2^2 S_{pq} \\ &s.t. \quad \mathbf{S} = \mathbf{S}^T, \quad \mathbf{S} \ge \mathbf{0}, \end{aligned} \tag{8}$$

where $\mathbf{y}_p$ and $\mathbf{y}_q$ are the pth and qth row vectors in $\mathbf{Y}$, respectively. We first remove the symmetric constraint and decouple it into $n \times n$ independent subproblems:

$$\begin{aligned} &\min_{S_{pq}} \sum_{v=1}^{V} \alpha_v^2 W_{pq}^2 \left(S_{pq} - \left(\hat{S}_{pq}^v - E_{pq}^v - E_{qp}^v \right) \right)^2 + \rho \left\| \mathbf{y}_p - \mathbf{y}_q \right\|_2^2 S_{pq} \\ &s.t. \quad S_{pq} \ge 0. \end{aligned} \tag{9}$$

Let $B_{pq} = \frac{2W_{pq}^2 \sum_{v=1}^{V} \alpha_v^2 (\hat{S}_{pq}^v - E_{pq}^v - E_{qp}^v) - \rho \left\| \mathbf{y}_p - \mathbf{y}_q \right\|_2^2}{2W_{pq}^2 \sum_{v=1}^{V} \alpha_v^2}$, we rewrite Eq. (9) as:

$$\begin{aligned} &\min_{S_{pq}} \left(S_{pq} - B_{pq} \right)^2 \\ &s.t. \quad S_{pq} \ge 0. \end{aligned}$$

The closed-form solution is

$$S_{pq} = \max(0, B_{pq}) \tag{10}$$

Notice that $\hat{\mathbf{S}}^v$, $\mathbf{E}^v + \mathbf{E}^{vT}$, and $\mathbf{W}$ are all symmetric. Therefore $\mathbf{B}$ is also symmetric and thus $\mathbf{S}$ learned from Eq. (10) satisfies the symmetric constraint.

3.2.2. Optimizing $\mathbf{E}^v$

When fixing $\mathbf{S}$, $\mathbf{Y}$ and α_v, we have:

$$\begin{aligned} &\min_{\mathbf{E}^v} \left\| \mathbf{W} \odot \left(\mathbf{S} - \left(\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \right) \right) \right\|_F^2 + \lambda \|\mathbf{E}^v\|_{2,1} \\ &s.t. \quad \hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \ge \mathbf{0}. \end{aligned} \tag{11}$$

For simplicity, We first ignore the constraint, and after obtaining the solution, we will prove that the obtained solution can satisfy this constraint. To handle the $l_{2,1}$-norm, following [41], we introduce a diagonal matrix $\mathbf{D}^v$, where ith diagonal element is $\frac{1}{2\left\|\mathbf{E}_i^v\right\|_2}$.

Setting the partial derivative of Eq. (11) w.r.t $\mathbf{E}^v$ to zero, we get:

$$2\mathbf{W}^2 \odot \left(\mathbf{E}^v + \mathbf{E}^{vT} \right) + \lambda \mathbf{D}^v \mathbf{E}^v = 2\mathbf{W}^2 \odot (\mathbf{S}^v - \mathbf{S}) \tag{12}$$

Let $\mathbf{F}^v = 2\mathbf{W}^2 \odot \left(\hat{\mathbf{S}}^v - \mathbf{S} \right)$. Considering pairwise elements E_{pq}^v and E_{qp}^v in Eq. (12), we obtain:

$$\begin{cases} 2W_{pq}^2 E_{pq}^2 + 2W_{pq}^2 E_{qp}^2 + \lambda D_{pp}^v E_{pq}^v = F_{pq}^v \\ 2W_{qp}^2 E_{qp}^v + 2W_{qp}^2 E_{pq}^v + \lambda D_{qq}^v E_{qp}^v = F_{qp}^v \end{cases} \tag{13}$$

Then, we get the solution of Eq. (13):

$$\begin{cases} E_{pq}^v = \frac{D_{qq}^v F_{pq}^v}{2W_{pq}^2 D_{qq}^v + 2W_{qp}^2 D_{pp}^v + \lambda D_{pp} D_{qq}} \\ E_{qp}^v = \frac{D_{pp}^v F_{qp}^v}{2W_{qp}^2 D_{pp}^v + 2W_{pq}^2 D_{qq}^v + \lambda D_{pp} D_{qq}} \end{cases}$$

It is easy to verify that both $\mathbf{F}^v$ and $\mathbf{W}$ are symmetric. Thus, E_{pq}^v can be simplified as the following form:

$$E_{pq}^v = \frac{D_{qq}^v F_{pq}^v}{2W_{pq}^2 (D_{pp}^v + D_{qq}^v) + \lambda D_{pp}^v D_{qq}^v} \tag{14}$$

Now, we show that the obtained solution satisfies the constraint:

$$\begin{aligned} & \hat{S}_{pq}^v - E_{pq}^v - E_{qp}^v \\ = \ & \hat{S}_{pq}^v - \frac{D_{qq}^v F_{pq}^v + D_{pp}^v F_{qp}^v}{2W_{pq}^2 D_{qq}^v + 2W_{qp}^2 D_{pp}^v + \lambda D_{pp} D_{qq}} \\ = \ & \frac{\lambda D_{pp} D_{qq} \hat{S}_{pq}^v + 2W_{pq}^2 (D_{pp}^v + D_{qq}^v) S_{pq}}{2W_{pq}^2 D_{qq}^v + 2W_{qp}^2 D_{pp}^v + \lambda D_{pp} D_{qq}} \\ \ge \ & 0 \end{aligned}$$

It is easy to verify that the obtained solution above satisfies the constraint because $\hat{S}_{pq}^v$, S_{pq}, and $\mathbf{D}^v$ are non-negative.

3.2.3. Optimizing Y

When fixing $\mathbf{S}$, α and $\mathbf{E}^v$, the optimization problem w.r.t. $\mathbf{Y}$ is:

$$\begin{aligned} &\min_{\mathbf{Y}} \quad tr(\mathbf{Y}^T \mathbf{L}\mathbf{Y}) \\ &s.t. \quad \mathbf{Y}^T\mathbf{Y} = \mathbf{I}. \end{aligned} \tag{15}$$

According to Ky Fan's theorem [40], we compute the eigen-decomposition of $\mathbf{L}$. Then we obtain the closed-form solution, which is consisted of the c eigenvectors of $\mathbf{L}$ corresponding to the c smallest eigenvalues.

3.2.4. Optimizing α_v

When fixing $\mathbf{S}$, $\mathbf{Y}$ and $\mathbf{E}^v$, we obtain α_v by solving the following problem:

$$\begin{aligned} &\min_{\alpha_v} \quad \sum_{v=1}^{V} \alpha_v^2 o_v \\ &s.t. \quad 0 \le \alpha_v \le 1, \quad \sum_{v=1}^{V} \alpha_v = 1, \end{aligned} \tag{16}$$

where $o_v = \left\| \mathbf{W} \odot \left(\mathbf{S} - \left(\hat{\mathbf{S}}^v - \mathbf{E}^v - \mathbf{E}^{vT} \right) \right) \right\|_F^2 + \lambda \|\mathbf{E}^v\|_{2,1}$. According to Cauchy–Schwarz inequality, we can obtain the closed-form solution of Eq. (16):

$$\alpha_v = \frac{o_v^{-1}}{\sum_{v=1}^{V} o_v^{-1}} \tag{17}$$

The whole algorithm to optimize Eq. (7) is summarized in Algorithm 1. When optimizing $\mathbf{S}$, $\mathbf{Y}$, and α_v, we can obtain the closed-form solution of the subproblems. When optimizing $\mathbf{E}^v$, according to [41], the objective function can also decrease monotonously. Moreover, the objective function has a lower bound. Therefore, Algorithm 1 always converges. In fact, Algorithm often converges very fast in practice.

3.2.5. Consistency score

Then, we apply the outlier matrix $\mathbf{E}^v$ to compute the consistency score. Intuitively, since $\mathbf{E}^v$ characterizes the class outliers on the graph, we can detect the outliers by finding the instances which have large $\mathbf{E}_{i.}^v$. Formally, we calculate the consistency score by:

$$s_i^c = \sum_{v=1}^{V} \|\mathbf{E}_{i.}^v\|_2^2. \tag{18}$$

For a class outlier $\mathbf{x}_i^v$, it behaviors inconsistently across multiple views, which means $\sum_{v=1}^{V} \left\|\mathbf{E}_{i.}^v\right\|_2^2$ is large, leading to large value of s_i^c. Notice

Algorithm 1 Optimization of Eq. (7).

```
Input: $\{\hat{\mathbf{S}}^v\}_{v=1}^V$, weight matrix $\mathbf{W}$.
Output: consensus matrix $\mathbf{S}$.
1: Initial $\mathbf{S} = \frac{1}{V}\sum_{v=1}^V \hat{\mathbf{S}}^v$, $\alpha_v = \frac{1}{V}$ and $\mathbf{E}^v = \frac{1}{2}\left(\hat{\mathbf{S}}^v - \mathbf{S}\right)$.
2: Construct Laplacian matrix $\mathbf{L}$, initialize $\mathbf{Y}$ by solving Eq. (15).
3: while not converge do
4:     Compute $\alpha$ by Eq. (17).
5:     Compute $\mathbf{S}$ by Eq. (10).
6:     Compute $\mathbf{E}^v$ by Eq. (14).
7:     Compute $\mathbf{Y}$ by solving Eq. (15).
8:     Adjust $\rho$ automatically.
9: end while
```

that the larger s_i^c is, the more likely $\mathbf{x}_i$ is a class attribute, which behaves opposite to the neighborhood score $\mathbf{s}^n$. To tackle this problem, we impose a linear transformation on $\mathbf{s}^c$ to reverse it, which maps the maximum of the score to 0 and maps the minimum of the score to 1. Denoting s_{max}^c as the maximum of the score and s_{min}^c as the minimum of the score, we calculate the new consistency score as:

$$s_i^c = \frac{s_{max}^c - s_i^c}{s_{max}^c - s_{min}^c}. \tag{19}$$

We use this $\mathbf{s}^c$ obtained by Eqs. (18) and (19) as the final consistency score.

3.3. Outlier scoring

Now, we introduce our outlier measurement criterion. As mentioned above, we calculate two score vectors $\mathbf{s}^n$ and $\mathbf{s}^c$ to detect attribute outliers and class outliers, respectively. Then, we define the final score $\mathbf{s} \in \mathbb{R}^n$ by combining them:

$$\mathbf{s} = \beta\mathbf{s}^n + (1-\beta)\mathbf{s}^c. \tag{20}$$

where $\beta \in (0, 1)$ is a balancing parameter. The smaller s_i is, the more likely it is an outlier. This outlier measurement criterion makes our method able to handle all three types of outliers simultaneously. We give the following explanations:

- For a normal instance, it is close to its neighbors, which makes the value of s_i^n large. On the other hand, since the normal instance is consistent across multiple views, its representation in the graph of each view is also close to the one in the consensus graph. Thus, the value of s_i^c should also be large. Consequently, its s_i should also be large.
- For an attribute outlier, it is obvious that the outlier is dissimilar to the majority of instances in each view. Thus, its value of s_i^n is small, leading to that its final score s_i will be smaller than that of normal instances.
- For a class outlier, it behaves inconsistently across different views, the difference between basic graphs and the consensus one should be large, which may introduce a large $\mathbf{E}_{i,\cdot}^v$, leading to a small value of s_i^c. Consequently, its final score s_i will also be smaller than that of normal instances.
- For a class-attribute outlier, since it is the mix of class outlier and attribute outlier, both s_i^n and s_i^c will be small, making the value of final score s_i small.

Thus, all three types of outliers will be detected by observing $\mathbf{s}$. We summarize our method in Algorithm 2.

3.4. Parameter setting

In this subsection, we introduce how to set the parameter in our method. We set t in Eq. (1) as the median of the Euclidean distances

Algorithm 2 MODGD Algorithm.

```
Input: Multi-view data set $\mathcal{X} = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^V\}$, hyper-parameter $\beta$.
Output: Outlier score $\mathbf{s}$.
1: Construct KNN graphs by Eq. (1).
2: Compute $\mathbf{s}^n$, $\mathbf{W}$ by Eqs. (2) and (5).
3: Compute $\{\hat{\mathbf{S}}^v\}_{v=1}^V$ by Eq. (3).
4: Optimize Eq. (7) by Algorithm 1.
5: Compute $\mathbf{s}^c$ by Eq. (18) and Eq. (19).
6: Compute $\mathbf{s}$ by Eq. (20).
```

among all data. When constructing the KNN graphs, we fix $k = 25$. We fix $c = 5$ and $\lambda = 0.1$. For ρ, we initialize $\rho = 1$ and adjust it by observing the rank of $\mathbf{L}$. In detail, if $rank(\mathbf{L}) > n\text{-}c$, which means the constraint is a little too weak, we increase $\rho \leftarrow 2 * \rho$; And if $rank(\mathbf{L}) < n\text{-}c$, we decrease it by $\rho \leftarrow 0.5 * \rho$. To get the final score, we tune β in $\{0.1, 0.2, \ldots, 0.9\}$.

3.5. Complexity analysis

Since we need to construct KNN graphs $\{\hat{\mathbf{S}}^v\}_{v=1}^V$, the time complexity is $O(n^2kV + n^2dV)$. In each iteration, when updating $\mathbf{E}^v$, the complexity is $O(n^2V)$ due to element-wise optimizations. When updating α, we compute o in $O(n^2V)$. For $\mathbf{S}$, we first compute $\mathbf{B}$, thus it costs $O(n^2c + n^2V)$. Since updating $\mathbf{Y}$ involves an eigenvector decomposition, it costs $O(n^2c)$. Supposing the number of iterations is T, the whole time complexity is $O\left(T\left(n^2V + n^2c\right) + n^2kV + n^2dV\right)$.

4. Experiment

In this section, we compare our method with some state-of-the-art multi-view outlier detection methods and evaluate it on benchmark data sets.

4.1. Toy data set

To show the effectiveness of the proposed method, we first show some results on a toy data set. This data contains 200 instances and 2 views, with each view having a feature dimension of 2, which is shown in Fig. 3. Figs. 3(a) and (b) show the two views of the data, where red triangles and red squares denote the two classes of the normal instances, respectively. The blue stars, diamonds, and dots denote the attribute outliers, class outliers, and class-attribute outliers, respectively.

Fig. 3(c) and (d) show the outlier scores learned by our method. The color represents the score. The red color means the score is high and the blue color means the score is low. It can be seen that most normal instances obtain high scores and most outliers obtain low scores in our method, which means our method correctly detects most of all three types of outliers.

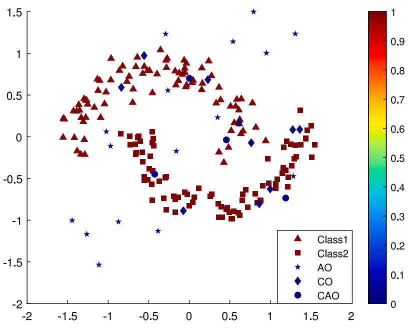
4.2. Real data sets

We also conduct experiments on 6 widely used real-world multi-view data sets, including 20newsgroup,[1] YaleB [42], Coil20 [43], Handwritten,[2] LandUse-21 [44], and Caltech101.[3] We summarize the detailed information of these data sets in Table 2.

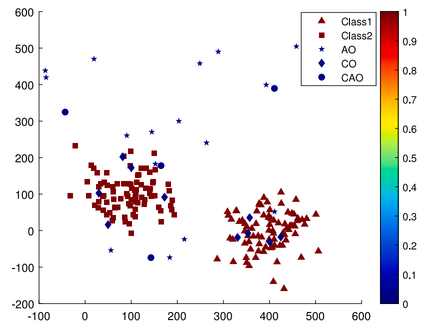
4.3. Experimental setup

Following the setting of [23], we generate three types of outliers on these data sets. Specifically, for each data set, we add 6 different

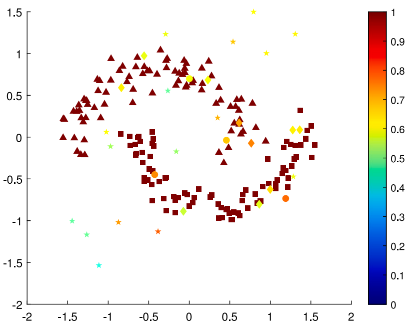
[1] https://lig-membres.imag.fr/grimal/data.html
[2] https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[3] http://www.vision.caltech.edu/datasets/



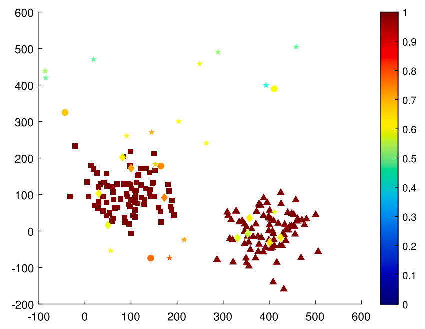
(a) Original data of the first view



(b) Original data of the second view



(c) Results of the first view



(d) Results of the second view

Fig. 3. Results on the toy data.

Table 2
Description of the data sets.

	#instances	#views	Features	#classes
20newsgroup	500	3	2000:2000:2000	5
YaleB	650	3	2500:3304:6750	10
Coil20	1440	3	1024:3304:6750	20
Handwritten	2000	6	216:76:64:6:240:47	10
LandUse-21	2100	3	20:59:40	21
Caltech101	9144	6	48:40:254:1984:512:680:32	102

Table 3
The settings of outlier ratios for different *id*.

id	D1	D2	D3	D4	D5	D6
ρ_1	0.02	0.02	0.05	0.05	0.08	0.08
ρ_2	0.05	0.08	0.02	0.08	0.02	0.05
ρ_3	0.08	0.05	0.08	0.02	0.05	0.02

ratios of outliers to generate six corrupted data sets, which are named "D + *id*". The *id* indicates the ratio of outliers attribute outlier (ρ_1), class outlier (ρ_2), and class-attribute outlier (ρ_3) contained in the data set, whose detailed information is shown in Table 3. For example, in Table 3, D1 means that 2% of the instances are attribute outliers, 5% of the instances are class outliers, and 8% of the instances are class-attribute outliers. The other 85% instances are normal instances.

We compare our method with the following unsupervised multiview outlier detection methods:

- HOAD [26]. It is a spectral clustering based outlier detection method.
- APOD [27]. It is an outlier detection method based on affinity propagation clustering.
- DMOD [25]. It is an outlier detection method based on k-means. It uses pair-wise constraints between two views to align all representations of data.
- CRMOD [28]. It applies k-means to learn consensus representations of data from all views for outlier detection.
- MLRA [29]. It is based on low-rank subspace clustering and uses pair-wise constraints to align all representations of data.
- LDSR [24]. It is similar to MLRA, but it separates the representations of different views into a consensus part and a view-specific part.
- MUVAD-FSR [30]. It estimates the set of normal instances based on the nearest neighbor-based outlier measurement criterion, and solves it with a fast spectral relaxation approach.
- MUVAD-QPR [30]. It solves the problem of MUVAD with a quadratic programming relaxation approach.
- NCMOD [23]. It is an outlier detection method based on a neighborhood consensus network.
- SRLSP [31]. It is a self-representation method for outlier detection, which preserves local similarity.

We use Area Under Curve (AUC) which is widely used in outlier detection tasks, to evaluate the results. We repeat the experiments 5 times and report the average results and standard deviations.

4.4. *Experiment results*

Tables 4–9 show the AUC results of our method and other compared multi-view outlier detection methods on each data set, respectively. We report the average results and the standard deviation of AUC. The best results of methods are in **boldface**. From Tables 4–9, we have the following points:

- Compared with the state-of-the-art methods based on the local similarity (i.e., MUVAD, NCMOD, and SRLSP), our method performs better on most data sets. Notice that NCMOD is the most recent deep multi-view outlier detection method, while ours is a shallow model and has a much simpler model. It well

Table 4
Detection performance on 20newsgroup.

Data	APOD [27]	CRMOD [28]	DMOD [25]	HOAD [26]	LDSR [24]	MLRA [29]	MUVAD-FSR [30]	MUVAD-QPR [30]	NCMOD [23]	SRLSP [31]	OURS
D1	0.2449± 0.0085	0.2067± 0.0097	0.3102± 0.2014	0.4903± 0.0262	0.5912± 0.0239	0.3984± 0.1634	0.6863± 0.0406	0.6982± 0.0382	0.8037± 0.0294	0.8371± 0.0234	**0.8947± 0.0063**
D2	0.3790± 0.0155	0.3239± 0.0367	0.3521± 0.1962	0.4870± 0.0079	0.5314± 0.0159	0.4207± 0.0833	0.6389± 0.0506	0.6489± 0.0481	0.7689± 0.0423	0.7655± 0.0281	**0.8419± 0.0086**
D3	0.0961± 0.0108	0.0915± 0.0079	0.5473± 0.4251	0.5895± 0.0430	0.4781± 0.0274	0.4459± 0.1128	0.7487± 0.0371	0.7601± 0.0378	0.8773± 0.0397	0.9170± 0.0139	**0.9538± 0.0119**
D4	0.3967± 0.0226	0.3304± 0.0141	0.2721± 0.0349	0.5834± 0.0376	0.3429± 0.0340	0.5056± 0.0134	0.6822± 0.0260	0.6811± 0.0269	0.7375± 0.0251	0.7352± 0.0159	**0.8227± 0.0311**
D5	0.0903± 0.0102	0.0872± 0.0127	0.2245± 0.3357	0.6791± 0.0729	0.3086± 0.0292	0.5832± 0.0259	0.8049± 0.0415	0.8226± 0.0370	0.9176± 0.0239	0.9304± 0.0130	**0.9530± 0.0110**
D6	0.2193± 0.0087	0.2085± 0.0082	0.1691± 0.0188	0.7063± 0.0238	0.2551± 0.0157	0.5663± 0.0431	0.7320± 0.0142	0.7448± 0.0106	0.8432± 0.0145	0.8372± 0.0141	**0.8884± 0.0253**

Table 5
Detection performance on YaleB.

Data	APOD [27]	CRMOD [28]	DMOD [25]	HOAD [26]	LDSR [24]	MLRA [29]	MUVAD-FSR [30]	MUVAD-QPR [30]	NCMOD [23]	SRLSP [31]	OURS
D1	0.7881± 0.0318	0.7887± 0.0235	0.7289± 0.1150	0.0597± 0.0103	0.2967± 0.0049	0.2613± 0.0451	0.4979± 0.0264	0.5000± 0.0000	0.7924± 0.0289	0.8695± 0.0216	**0.9793± 0.0049**
D2	0.7513± 0.0245	0.7662± 0.0255	0.7076± 0.0797	0.0775± 0.0059	0.4797± 0.0141	0.3684± 0.0189	0.5086± 0.0277	0.5000± 0.0000	0.7106± 0.0218	0.7887± 0.0302	**0.9593± 0.0236**
D3	0.7570± 0.0499	0.8020± 0.0145	0.6081± 0.3000	0.1658± 0.0152	0.1062± 0.0122	0.1223± 0.0350	0.5068± 0.0253	0.5000± 0.0000	0.8799± 0.0447	0.9530± 0.0079	**0.9961± 0.0036**
D4	0.6995± 0.0332	0.7448± 0.0255	0.6445± 0.2018	0.1677± 0.0195	0.4721± 0.0131	0.4328± 0.1063	0.4818± 0.0429	0.5000± 0.0000	0.7541± 0.0226	0.7996± 0.0239	**0.9640± 0.0109**
D5	0.6583± 0.0338	0.7846± 0.0162	0.7229± 0.1957	0.2494± 0.0264	0.1037± 0.0130	0.4001± 0.1649	0.4632± 0.0211	0.5000± 0.0000	0.9262± 0.0246	0.9552± 0.0078	**0.9922± 0.0098**
D6	0.6494± 0.0294	0.7624± 0.0201	0.6676± 0.2069	0.2696± 0.0334	0.2686± 0.0172	0.5246± 0.1024	0.4799± 0.0321	0.5000± 0.0000	0.8127± 0.0833	0.8819± 0.0137	**0.9796± 0.0086**

Table 6
Detection performance on Coil20.

Data	APOD [27]	CRMOD [28]	DMOD [25]	HOAD [26]	LDSR [24]	MLRA [29]	MUVAD-FSR [30]	MUVAD-QPR [30]	NCMOD [23]	SRLSP [31]	OURS
D1	0.6627± 0.0175	0.7978± 0.0209	0.7022± 0.1884	0.0098± 0.0019	0.3522± 0.0058	0.3891± 0.0874	0.5386± 0.0125	0.5542± 0.0067	0.8589± 0.0218	0.9486± 0.0102	**0.9980± 0.0012**
D2	0.6924± 0.0212	0.7746± 0.0170	0.7202± 0.1194	0.0062± 0.0031	0.5469± 0.0045	0.3874± 0.0861	0.5340± 0.0198	0.5331± 0.0089	0.7844± 0.0118	0.8858± 0.0144	**0.9992± 0.0003**
D3	0.5185± 0.0287	0.7758± 0.0490	0.5538± 0.2598	0.0347± 0.0063	0.1495± 0.0086	0.3274± 0.1036	0.5624± 0.0162	0.5754± 0.0056	0.9449± 0.0015	0.9735± 0.0052	**0.9984± 0.0021**
D4	0.5723± 0.0281	0.7319± 0.0241	0.6507± 0.1381	0.0499± 0.0090	0.5269± 0.0025	0.5205± 0.1200	0.5349± 0.0220	0.5351± 0.0097	0.7658± 0.0192	0.9015± 0.0132	**0.9967± 0.0038**
D5	0.3900± 0.0120	0.7330± 0.0194	0.6563± 0.2468	0.0585± 0.0154	0.1385± 0.0048	0.3496± 0.1447	0.5671± 0.0143	0.5839± 0.0048	0.9453± 0.0127	0.9712± 0.0090	**0.9996± 0.0002**
D6	0.4320± 0.0078	0.7237± 0.0157	0.6050± 0.1667	0.1081± 0.0124	0.3330± 0.0017	0.6409± 0.0813	0.5609± 0.0258	0.5601± 0.0052	0.8479± 0.0138	0.9383± 0.0126	**0.9991± 0.0010**

Table 7
Detection performance on Handwritten.

Data	APOD [27]	CRMOD [28]	DMOD [25]	HOAD [26]	LDSR [24]	MLRA [29]	MUVAD-FSR [30]	MUVAD-QPR [30]	NCMOD [23]	SRLSP [31]	OURS
D1	0.8543± 0.0110	0.9254± 0.0063	0.5623± 0.2153	0.1274± 0.0068	0.9766± 0.0075	0.4302± 0.0418	0.5591± 0.0156	0.3665± 0.0158	0.8278± 0.0063	0.9325± 0.0126	**0.9832± 0.0074**
D2	0.8126± 0.0097	0.8769± 0.0103	0.5800± 0.1287	0.1403± 0.0031	0.9477± 0.0131	0.4221± 0.0227	0.5441± 0.0065	0.3850± 0.0165	0.7471± 0.0072	0.8993± 0.0054	**0.9720± 0.0058**
D3	0.8857± 0.0103	0.9692± 0.0031	0.4337± 0.3286	0.2298± 0.0037	0.9924± 0.0055	0.4871± 0.0507	0.5867± 0.0164	0.3624± 0.0209	0.9253± 0.0046	0.9676± 0.0101	**0.9932± 0.0035**
D4	0.8277± 0.0079	0.8674± 0.0104	0.6914± 0.0904	0.2540± 0.0047	0.9464± 0.0102	0.4699± 0.0123	0.5617± 0.0072	0.3947± 0.0390	0.7499± 0.0151	0.8987± 0.0213	**0.9732± 0.0052**
D5	0.9061± 0.0098	0.9665± 0.0096	0.5342± 0.2346	0.3254± 0.0074	0.9885± 0.0049	0.5662± 0.0130	0.5940± 0.0126	0.3583± 0.0186	0.9254± 0.0071	0.9723± 0.0059	**0.9924± 0.0040**
D6	0.8557± 0.0185	0.9182± 0.0068	0.7773± 0.1265	0.3447± 0.0104	0.9697± 0.0083	0.5512± 0.0256	0.5830± 0.0151	0.3662± 0.0072	0.8400± 0.0122	0.9333± 0.0187	**0.9833± 0.0027**

Table 8
Detection performance on Landuse-21.

Data	APOD [27]	CRMOD [28]	DMOD [25]	HOAD [26]	LDSR [24]	MLRA [29]	MUVAD-FSR [30]	MUVAD-QPR [30]	NCMOD [23]	SRLSP [31]	OURS
D1	0.6350± 0.0319	0.8844± 0.0092	0.7646± 0.1530	0.1722± 0.0105	0.9565± 0.0071	0.4937± 0.0158	0.7156± 0.0261	0.7283± 0.0229	0.8955± 0.0089	0.9474± 0.0030	**0.9622± 0.0049**
D2	0.6162± 0.0211	0.8047± 0.0146	0.7051± 0.1512	0.2010± 0.0081	0.9137± 0.0048	0.5166± 0.0238	0.6495± 0.0066	0.6645± 0.0058	0.8390± 0.0154	0.9072± 0.0075	**0.9282± 0.0109**
D3	0.5810± 0.0172	0.9426± 0.0128	0.6101± 0.3420	0.3453± 0.0058	0.9834± 0.0058	0.4697± 0.0195	0.8207± 0.0192	0.8340± 0.0192	0.9331± 0.0241	0.9799± 0.0083	**0.9806± 0.0055**
D4	0.5760± 0.0356	0.8332± 0.0093	0.7227± 0.1745	0.3837± 0.0085	0.9278± 0.0074	0.5202± 0.0279	0.6966± 0.0185	0.7104± 0.0172	0.8638± 0.0123	0.9194± 0.0082	**0.9371± 0.0111**
D5	0.5276± 0.0415	0.9416± 0.0075	0.9314± 0.0072	0.5382± 0.0057	0.9830± 0.0036	0.4597± 0.0236	0.8629± 0.0066	0.8755± 0.0066	0.9276± 0.0146	0.9773± 0.0026	**0.9810± 0.0036**
D6	0.5550± 0.0395	0.8891± 0.0036	0.8631± 0.0137	0.5617± 0.0101	0.9589± 0.0083	0.4970± 0.0253	0.7862± 0.0136	0.8009± 0.0119	0.9047± 0.0052	0.9463± 0.0057	**0.9619± 0.0069**

Table 9
Detection performance on Caltech101. MLRA and SRLSP cannot run a result in a reasonable time on the large data set Caltech101.

Data	APOD [27]	CRMOD [28]	DMOD [25]	HOAD [26]	MUVAD-FSR [30]	MUVAD-QPR [30]	NCMOD [23]	SRLSP [31]	OURS
D1	0.5146± 0.0108	0.8693± 0.0108	0.5397± 0.2960	0.2362± 0.0056	0.5013± 0.0077	0.5017± 0.0006	0.8800± 0.0058	0.8212± 0.0067	**0.9041± 0.0072**
D2	0.5201± 0.0119	0.8125± 0.0109	0.7905± 0.0071	0.2662± 0.0046	0.4980± 0.0077	0.5009± 0.0004	0.8093± 0.0056	0.7466± 0.0087	**0.8485± 0.0073**
D3	0.4209± 0.0172	0.9314± 0.0148	0.8080± 0.2697	0.3660± 0.0023	0.5042± 0.0079	0.5023± 0.0006	0.9422± 0.0019	0.9170± 0.0053	**0.9586± 0.0018**
D4	0.3899± 0.0090	0.8172± 0.0135	0.7903± 0.0060	0.4117± 0.0075	0.5023± 0.0078	0.5009± 0.0011	0.8165± 0.0050	0.7458± 0.0070	**0.8418± 0.0091**
D5	0.3007± 0.0039	0.9291± 0.0210	0.7782± 0.3286	0.5302± 0.0045	0.4982± 0.0075	0.5031± 0.0010	0.9518± 0.0057	0.9246± 0.0035	**0.9611± 0.0027**
D6	0.2696± 0.0079	0.8788± 0.0116	0.8606± 0.0079	0.5591± 0.0064	0.4962± 0.0068	0.5025± 0.0007	0.8908± 0.0069	0.8448± 0.0058	**0.8977± 0.0014**

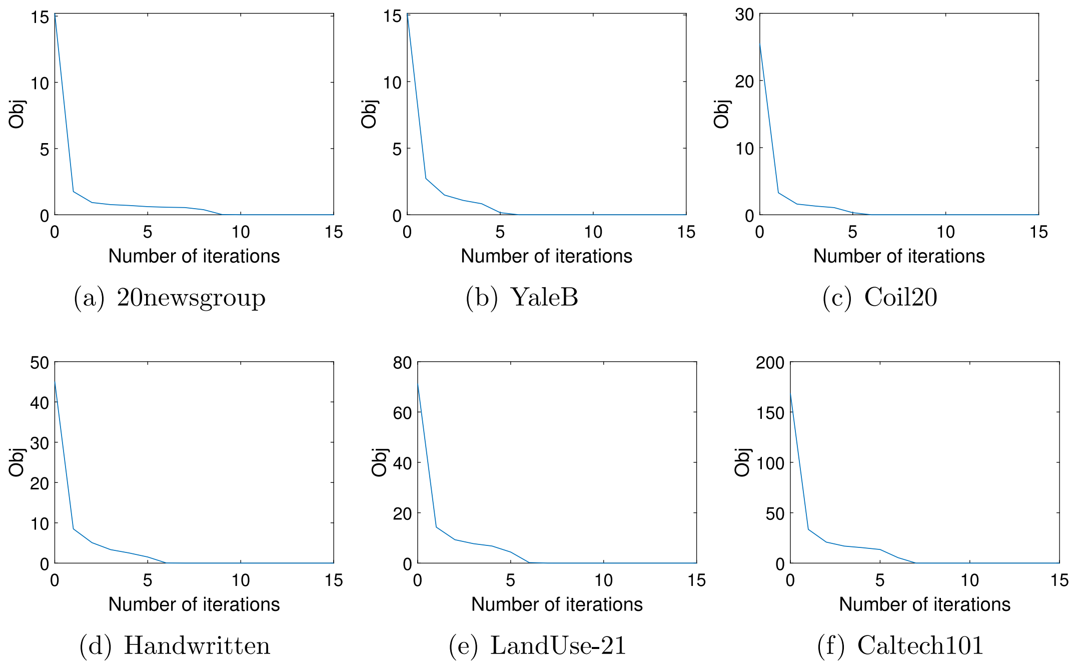


Fig. 4. Convergence curves of our method.

demonstrates the effectiveness and superiority of our method. The reason may be that our method can fully utilize the relation between data and explicitly characterize the structure of outliers when learning the consensus to detect class outliers.

- For pair-wise constraint based methods HOAD, APOD, DMOD, and MLRA, they often perform worse than other non pair-wise constraint based methods on most data sets, which illustrates the limitation of the pair-wise constraint. Since our method learns the consensus graph across all views, it has better performance than these methods on all data sets.
- Consider SRLSP, LDSR, and CRMOD, which obtain consensus information via a simple linear combination of all views. Our

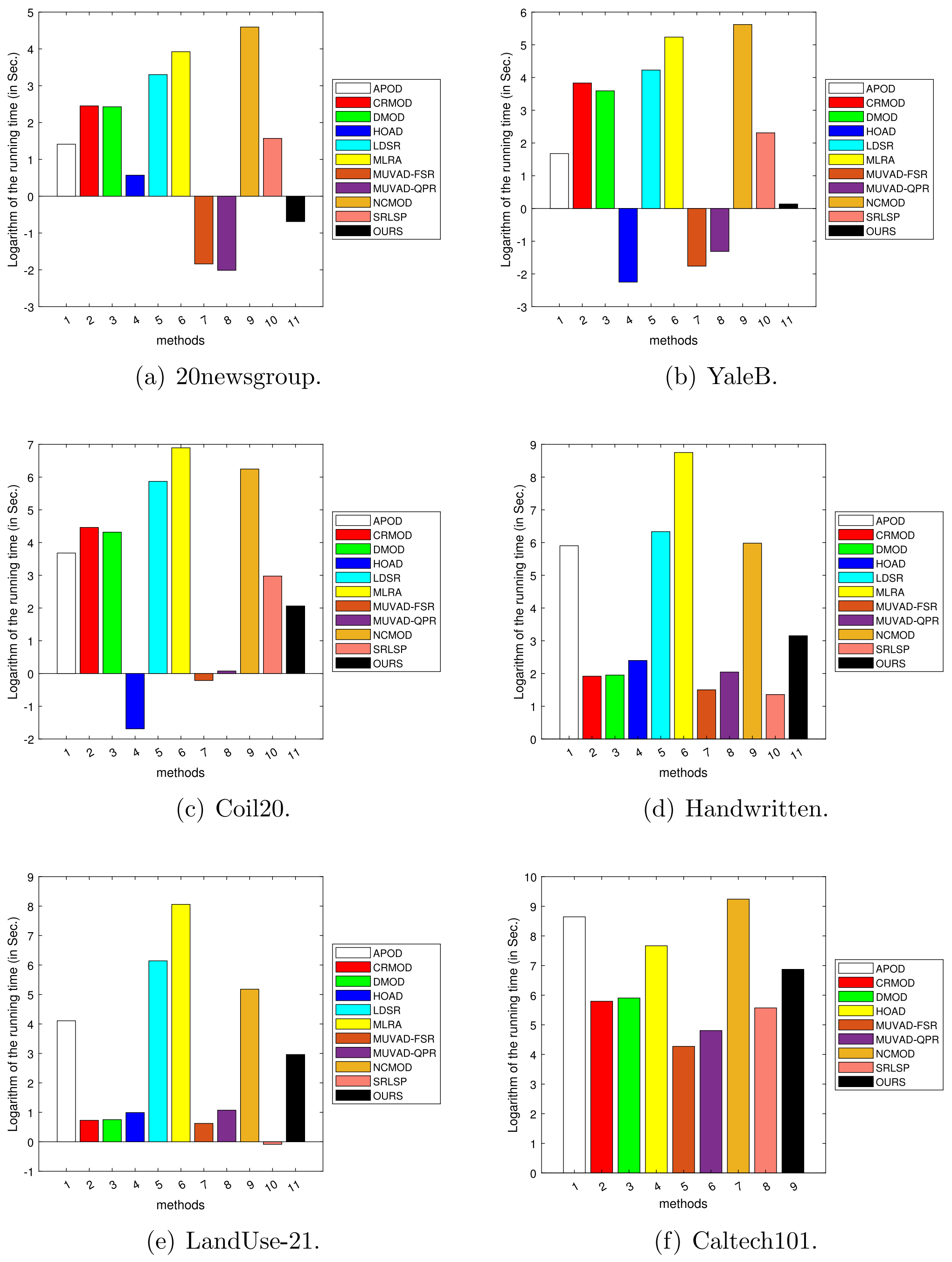


Fig. 5. Running time (in Sec.) of all methods. Notice that MLRA and LDSR cannot run a result in a reasonable time on Caltech101.

method also outperforms them. It well demonstrates that the ensemble method used in ours can capture the intrinsic structure of class outliers when learning the consensus graph more effectively.

4.5. *Efficiency results*

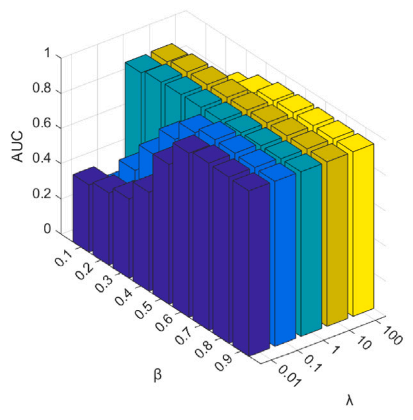
Fig. 4 shows the convergence curves of our method on all data sets. It can be seen that our method converges very fast (i.e., often converges within 10 iterations).

Fig. 5 shows the running time of all methods on all data sets. Due to the significant difference in the running time of some methods, for better comparison, we report the logarithm of the time in Fig. 5. Notice that LDSR and MLRA cannot run a result in a reasonable time in large data set Caltech101. It can be seen that the running time of our method is comparable with other state-of-the-art methods and it is even faster than some methods, such as APOD, LDSR, MLRA, and NCMOD.

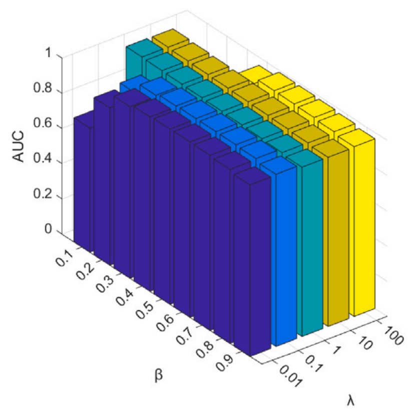
4.6. *Ablation study*

To show the effectiveness of our method, we compare our method with the following three degenerated versions on 20newsgroups and Handwritten:

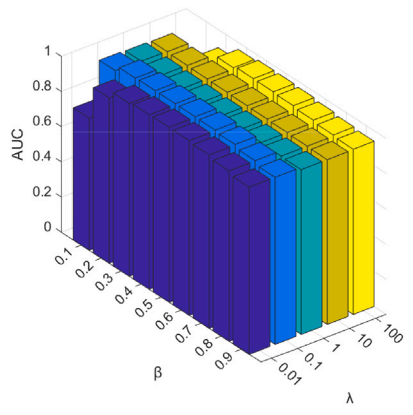
- $\mathbf{s}^n$. It only uses the neighborhood score to detect outliers.
- $\mathbf{s}^c$. Only the consistency score is regarded as the final score.
- $\mathbf{s}^n + \mathbf{s}^c_{ini}$. It uses the initial $\mathbf{E}^v$ to obtain the consistency score, then combine the neighborhood score to get the final score.



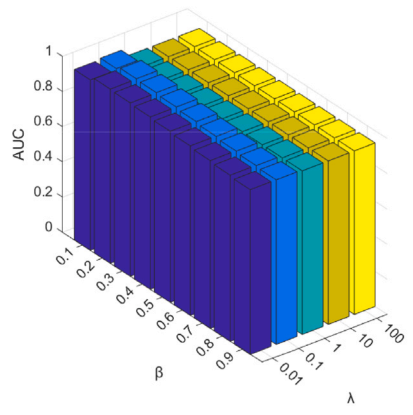
(a) 20newsgroups



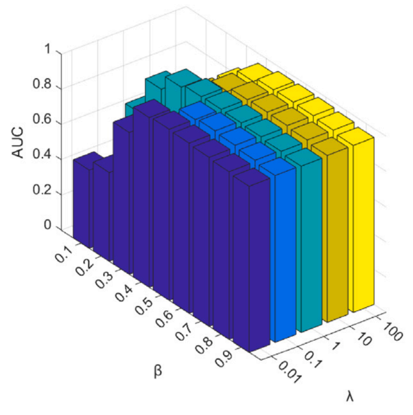
(b) YaleB



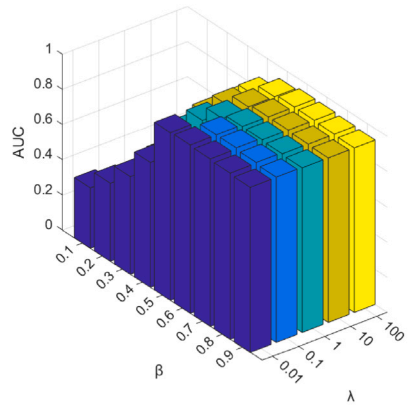
(c) Coil20



(d) Handwritten



(e) LandUse-21



(f) Caltech101

Fig. 6. AUC with respect to β and λ.

- $\mathbf{s}^n + \mathbf{s}^c$. It is the complete model of our method with all parts.

Tables 10, 11 show the results of the ablation study. The results on other data sets are similar. From these Tables, we find that using one of the two scores as the final score cannot effectively detect all types of outliers. In more detail, $\mathbf{s}^n$ can only effectively detect attribute outliers and $\mathbf{s}^c$ can only detect the class-attribute outliers. When the ratio of these two types of outliers is high (i.e., D3, D5), it can obtain a high value of AUC. Similarly, $\mathbf{s}^c$ has good performance when the ratio of class outliers is high (i.e., D2, D4). When we combine these two scores, it achieves better performance than using either of them alone. Our method outperforms $\mathbf{s}^n + \mathbf{s}^c_{ini}$, which illustrates the effectiveness of characterizing the structure of class outliers.

4.7. *Parameter study*

Our method contains two hyper-parameters β and λ. In our experiments, we fix λ to 0.1 and do not tune it. Despite this, to comprehensively show the effects of these two hyper-parameters, we show the results w.r.t. β in $\{0.1, 0.2, \dots, 0.9\}$ and λ in $\{0.01, 0.1, 1, 10, 100\}$ in Fig. 6.

Table 10
Compared with degenerated versions on 20newsgroups.

	D1	D2	D3	D4	D5	D6
$\mathbf{s}^c$	0.1960± 0.0311	0.3314± 0.0228	0.0904± 0.0262	0.3007± 0.0359	0.0826± 0.0143	0.1656± 0.0283
$\mathbf{s}^n$	0.8388± 0.0204	0.7219± 0.0228	0.9361± 0.0152	0.7439± 0.0367	0.9355± 0.0175	0.8617± 0.0234
$\mathbf{s}^n + \mathbf{s}^c_{ini}$	0.8508± 0.0142	0.7444± 0.0184	0.9372± 0.0112	0.7651± 0.0342	0.9385± 0.0185	0.8627± 0.0192
$\mathbf{s}^n + \mathbf{s}^c$	**0.8947± 0.0063**	**0.8419± 0.0086**	**0.9538± 0.0119**	**0.8227± 0.0311**	**0.9530± 0.0110**	**0.8884± 0.0253**

Table 11
Compared with degenerated versions on Handwritten.

	D1	D2	D3	D4	D5	D6
$\mathbf{s}^c$	0.7474± 0.0252	0.7919± 0.0051	0.5640± 0.0242	0.6215± 0.0120	0.3852± 0.0211	0.4223± 0.0145
$\mathbf{s}^n$	0.8336± 0.0049	0.7383± 0.0112	0.9312± 0.0046	0.7253± 0.0183	0.9310± 0.0061	0.8372± 0.0164
$\mathbf{s}^n + \mathbf{s}^c_{ini}$	0.9334± 0.0088	0.8872± 0.0097	0.9714± 0.0046	0.8805± 0.0124	0.9677± 0.0054	0.9277± 0.0117
$\mathbf{s}^n + \mathbf{s}^c$	**0.9832± 0.0074**	**0.9720± 0.0058**	**0.9932± 0.0035**	**0.9732± 0.0052**	**0.9924± 0.0040**	**0.9833± 0.0027**

It can be seen that our method is stable in a wide range. Specifically, we can select β from $[0.2, 0.7]$, which often obtains a relatively good performance.

5. Conclusion

In this paper, we proposed a novel unsupervised multi-view outlier detection method. To avoid making the assumption of clustering, we constructed multiple graphs to represent the structure of multi-view data. We detected the attribute outliers directly on the multiple graphs. Then, we learned a consensus graph to detect class outliers. When learning the consensus graph, we explicitly extracted the structured class outliers on graphs and recovered the multiple clean graphs for the ensemble. At last, we designed an outlier measurement criterion score by combining the attribute outlier score and class outlier score, which can detect all three types of outliers on multi-view data. Extensive experiments shew that the proposed method outperformed the state-of-the-art unsupervised multi-view outlier detection methods, which demonstrated the superiority and effectiveness of this method.

Although the proposed method performs well when detecting outliers, it has high time complexity because it needs to handle multiple graphs. In the future, we will try to reduce the time complexity further and apply it to some large-scale data sets.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the National Natural Science Foundation of China grants 62176001, 61806003, and 62376146. This work is also supported by the Natural Science Project of Anhui Provincial Education Department grants 2023AH030004.

References

[1] B. Schölkopf, R.C. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, Support vector method for novelty detection, in: Advances in Neural Information Processing Systems, Vol. 12, 1999.
[2] L. Xiong, X. Chen, J. Schneider, Direct robust matrix factorizatoin for anomaly detection, in: 2011 IEEE 11th International Conference on Data Mining, IEEE, 2011, pp. 844–853.
[3] T. De Vries, S. Chawla, M.E. Houle, Finding local anomalies in very high dimensional space, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 128–137.
[4] Z. He, X. Xu, S. Deng, Discovering cluster-based local outliers, Pattern Recognit. Lett. 24 (9–10) (2003) 1641–1650.
[5] M.-F. Jiang, S.-S. Tseng, C.-M. Su, Two-phase clustering process for outliers detection, Pattern Recognit. Lett. 22 (6–7) (2001) 691–700.
[6] M. Kontaki, A. Gounaris, A.N. Papadopoulos, K. Tsichlas, Y. Manolopoulos, Continuous monitoring of distance-based outliers over data streams, in: 2011 IEEE 27th International Conference on Data Engineering, IEEE, 2011, pp. 135–146.
[7] M. Salehi, C. Leckie, J.C. Bezdek, T. Vaithianathan, X. Zhang, Fast memory efficient local outlier detection in data streams, IEEE Trans. Knowl. Data Eng. 28 (12) (2016) 3246–3260.
[8] S. Sathe, C.C. Aggarwal, Subspace outlier detection in linear time with randomized hashing, in: 2016 IEEE 16th International Conference on Data Mining, ICDM, IEEE, 2016, pp. 459–468.
[9] 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.
[10] R. Yu, X. He, Y. Liu, Glad: Group anomaly detection in social media analysis, ACM Trans. Knowl. Discov. Data (TKDD) 10 (2) (2015) 1–22.
[11] R. Yu, H. Qiu, Z. Wen, C. Lin, Y. Liu, A survey on social media anomaly detection, ACM SIGKDD Explor. Newsl. 18 (1) (2016) 1–14.
[12] C.-D. Wang, Z.-H. Deng, J.-H. Lai, S.Y. Philip, Serendipitous recommendation in e-commerce using innovator-based collaborative filtering, IEEE Trans. Cybern. 49 (7) (2018) 2678–2692.
[13] A. Boukerche, L. Zheng, O. Alfandi, Outlier detection: Methods, models, and classification, ACM Comput. Surv. 53 (3) (2020) 1–37.
[14] J. Chen, S. Sathe, C. Aggarwal, D. Turaga, Outlier detection with autoencoder ensembles, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 90–98.
[15] G. Pang, L. Cao, L. Chen, H. Liu, Learning representations of ultrahigh-dimensional data for random distance-based outlier detection, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2041–2050.
[16] G. Pang, C. Shen, L. Cao, A.V.D. Hengel, Deep learning for anomaly detection: A review, ACM Comput. Surv. (CSUR) 54 (2) (2021) 1–38.
[17] G. Pang, C. Yan, C. Shen, A.v.d. Hengel, X. Bai, Self-trained deep ordinal regression for end-to-end video anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12173–12182.
[18] H. Wang, G. Pang, C. Shen, C. Ma, Unsupervised representation learning by predicting random distances, 2019, arXiv preprint arXiv:1912.12186.
[19] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: International Conference on Machine Learning, PMLR, 2016, pp. 478–487.
[20] B. Zong, Q. Song, M.R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.
[21] D.M. Hawkins, Identification of Outliers, Vol. 11, Springer, 1980.
[22] L. Ruff, R.A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, M. Kloft, Deep semi-supervised anomaly detection, 2019, arXiv preprint arXiv:1906.02694.
[23] L. Cheng, Y. Wang, X. Liu, Neighborhood consensus networks for unsupervised multi-view outlier detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 8, 2021, pp. 7099–7106.
[24] K. Li, S. Li, Z. Ding, W. Zhang, Y. Fu, Latent discriminant subspace representations for multi-view outlier detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1, 2018.
[25] H. Zhao, Y. Fu, Dual-regularized multi-view outlier detection, in: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI '15, AAAI Press, 2015, pp. 4077–4083.
[26] 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.
[27] A. Marcos Alvarez, M. Yamada, A. Kimura, T. Iwata, Clustering-based anomaly detection in multi-view data, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013, pp. 1545–1548.
[28] H. Zhao, H. Liu, Z. Ding, Y. Fu, Consensus regularized multi-view outlier detection, IEEE Trans. Image Process. 27 (1) (2017) 236–248.

[29] S. Li, M. Shao, Y. Fu, Multi-view low-rank analysis for outlier detection, in: Proceedings of the 2015 SIAM International Conference on Data Mining, SIAM, 2015, pp. 748–756.
[30] X.-R. Sheng, D.-C. Zhan, S. Lu, Y. Jiang, Multi-view anomaly detection: Neighborhood in locality matters, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01, 2019, pp. 4894–4901.
[31] Y. Wang, C. Chen, J. Lai, L. Fu, Y. Zhou, Z. Zheng, A self-representation method with local similarity preserving for fast multi-view outlier detection, ACM Trans. Knowl. Discov. Data 17 (1) (2023) 1–20.
[32] P. Zhou, L. Du, L. Shi, H. Wang, Y.-D. Shen, Recovery of corrupted multiple kernels for clustering, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
[33] P. Zhou, Y.-D. Shen, L. Du, F. Ye, Incremental multi-view support vector machine, in: Proceedings of the 2019 SIAM International Conference on Data Mining, SIAM, 2019, pp. 1–9.
[34] P. Zhou, Y.-D. Shen, L. Du, F. Ye, X. Li, Incremental multi-view spectral clustering, Knowl.-Based Syst. 174 (2019) 73–86.
[35] P. Zhou, X. Liu, L. Du, X. Li, Self-paced adaptive bipartite graph learning for consensus clustering, ACM Trans. Knowl. Discov. Data 17 (5) (2023) 62:1–62:35.
[36] P. Zhou, L. Du, Learnable graph filter for multi-view clustering, in: ACM International Conference on Multimedia, MM 2023, 2023.
[37] P. Zhou, L. Du, X. Li, Adaptive consensus clustering for multiple K-means via base results refining, IEEE Trans. Knowl. Data Eng. (2023) 1–14.
[38] H. Wang, F. Nie, H. Huang, Multi-view clustering and feature learning via structured sparsity, in: International Conference on Machine Learning, PMLR, 2013, pp. 352–360.
[39] 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, 2014, pp. 977–986.
[40] K. Fan, On a theorem of Weyl concerning eigenvalues of linear transformations I, Proc. Natl. Acad. Sci. 35 (11) (1949) 652–655.
[41] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, X. Zhou, $L_{2,1}$-norm regularized discriminative feature selection for unsupervised learning, in: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011, IJCAI/AAAI, 2011, pp. 1589–1594.
[42] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.
[43] S.A. Nene, S.K. Nayar, H. Murase, et al., Columbia Object Image Library (Coil-20), Citeseer, 1996.
[44] Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010, pp. 270–279.