Big Data Research 29 (2022) 100332

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

A Flexible Outlier Detector Based on a Topology Given by Graph Communities

Oriol Ramos Terrades *, Albert Berenguel, Débora Gil *

Computer Vision Center and the Department of Computer Science, Universitat Autònoma de Barcelona, Cerdanyola del Valles, 08193, Spain

ARTICLE INFO

Article history:
Received 12 October 2021
Received in revised form 8 June 2022
Accepted 5 July 2022
Available online 13 July 2022

Keywords:
Classification algorithms
Detection algorithms
Description of feature space local structure
Graph communities
Machine learning algorithms
Outlier detectors

ABSTRACT

Outlier detection is essential for optimal performance of machine learning methods and statistical predictive models. Their detection is especially determinant in small sample size unbalanced problems, since in such settings outliers become highly influential and significantly bias models. This particular experimental settings are usual in medical applications, like diagnosis of rare pathologies, outcome of experimental personalized treatments or pandemic emergencies. In contrast to population-based methods, neighborhood based local approaches compute an outlier score from the neighbors of each sample, are simple flexible methods that have the potential to perform well in small sample size unbalanced problems. A main concern of local approaches is the impact that the computation of each sample neighborhood has on the method performance. Most approaches use a distance in the feature space to define a single neighborhood that requires careful selection of several parameters, like the number of neighbors.

This work presents a local approach based on a local measure of the heterogeneity of sample labels in the feature space considered as a topological manifold. Topology is computed using the communities of a weighted graph codifying mutual nearest neighbors in the feature space. This way, we provide with a set of multiple neighborhoods able to describe the structure of complex spaces without parameter fine tuning. The extensive experiments on real-world and synthetic data sets show that our approach outperforms, both, local and global strategies in multi and single view settings.

1. Introduction

Outlier, or anomaly, detection is a major issue in many learning-based algorithms since the presence of outlier data on training data might affect the proper estimation of model parameters. Moreover, outlier detection becomes harder when data changes along time, since it is unclear how to distinguish proper data, coming from a new arising class, from corrupted, or outlier, data. Outlier detection is not just a technical step in a data cleaning process. It is, by itself, a key topic in many fields such as fraudulent document detection, in identity documents or passports, insurances claims and health care fraud; medical applications and assisted diagnosis systems; detection of security threats, etc. In all these tasks, an accurate outlier detection will impact in the economic balance of any company or to properly detect any security threat, for instance.

The outlier concept is fuzzy, it depends on each task and thus each detection method provides its own definition. The authors in [1] define an outlier as a sample that "does not conform to expected behavior" and they classify them as *point outliers*, *contextual outliers* and *collective outliers*. Point outliers are samples with abnormal feature values not expected for any of the classes and they correspond to isolated samples either not belonging to any cluster or not following the population distribution. Contextual outliers are samples which are labeled differently from their neighboring samples, which define their context. In recent works authors detect outliers comparing samples to the other samples in the same community [2,3]. Collective outliers are small group of samples sharing unusual features that are clustered together. For instance, the authors in [4] detect these type of outliers on evolutionary networks according to communities topological properties. In [5] split outliers are split into *attribute outliers* and *class outliers* in the context of multimodal representations. In such representations, usual in radiomics and computer assisted diagnosis [6], there are two or more feature spaces (called *views*) for each sample. Attribute outliers correspond to point outliers, while class outliers are similar to contextual outliers in the measure that are samples labeled dif-

* Corresponding authors.
E-mail addresses: oriolrt@cvc.uab.cat (O. Ramos Terrades), debora@cvc.uab.cat (D. Gil).

https://doi.org/10.1016/j.bdr.2022.100332

ferently across *views*. It follows that attribute and class outliers are different types of anomalous points and, thus, they are usually detected with different methods. In general existing methods only perform optimally in one of the two types [7].

In this work we present a local approach based on a measure of sample labels diversity in a set of topological neighborhoods of each sample. We compute this topology extending the graph-based method presented in [8]. In particular, we refine the initial topology given by community detection methods to include isolated non-outlier points. Sample diversity is computed using probabilistic measures, which summarize the variability of sample labels in the set of topological neighborhoods samples belong to. These diversity measures provide a normalized *outlierness* representation feature space. This normalized space is independent of the actual sample features and only depends on their topological structure. Finally, a classifier, like support vector machine or random forest, is used in a final step to detect outliers. We call our method Community-based Outlier Detector, COD.

All in all, we contribute to the detection of outliers in the following aspects:

- A local description of feature spaces topology based on the communities of a mutual-knn graph (mkNN) able to model complex data distributions.
- A normalized representation space of the degree of outlierness based on statistical measures of sample labels heterogeneity in the topological neighborhoods given by mkNN communities.

2. Related work

Existing methods for detection of outliers can be categorized into global, local, graph-based and multiview approaches. Global methods are population based and they model the distribution in the feature space of a set of (annotated) samples. Population distribution can be modeled using either parametric global descriptors or unsupervised clustering approaches. These methods are suited to detect point (or attribute) outliers as far as sample size is relatively large and balanced. Local methods are based on a description of the structure of each sample's neighbors in the feature space and, thus, they are better suited to detect, both, attribute and class outliers, even in small sample size problems. Moreover, outlier detector for structured data focus on identifying sub-graphs that broke repetitive patterns. Finally, outlier detectors for multiview data focus on finding samples that are not consistently labeled across views. Below, we review the main methods in each category.

2.1. Global methods

Global approaches seek to estimate parametric data distributions and define outliers as points not following the estimated distribution. For instance, in [9] the authors use Gaussian mixture models (GMM) for outlier detection in which they define each point as a cluster and the outlierness score is the reciprocal of the point likelihood. In general, data distribution cannot be properly estimated in small sample size sets since outliers become influential points which deviate global approaches from normal population. This leads to lack of reproducibility and drop their potential for outlier detection. Similarly, the authors in [10] propose a Hidden Markov Random Field (HMRF) on attributed graphs to compute node outliers. Hidden variables represent community labels, ranging from 0 to k, model parameters are estimated by the EM algorithm and partition functions on each node are estimated accordingly. Outliers are those nodes with partition function close to 0.

With the advent of big data and deep learning techniques this main drawback of global approaches seems to be mitigated since they can learn complex data structures from big amount of data. A main issue when dealing with big data is labeling enough samples for training and testing deep learning methods, which is especially difficult in such an imbalanced classification task. In [11], the authors define four groups of outlier detectors methods: unsupervised, semi-supervised, hybrid and One-Class neural networks. While is clear which methods belong to unsupervised and semi-supervised methods, hybrid methods use deep learning architectures for feature extraction and then use traditional outlier detectors. Finally, One-Class neural network (OC-NN) method is inspired by kernel methods on one-class classification tasks [12]. A variant of OC-NN architectures is Deep Support Vector Data Description (Deep-SVDD) [13]. In that work, the authors train a deep neural network to extract common variation factors by mapping close inlier data instances to the center of a hyper-sphere. Generative adversarial active learning (GAAL) networks are used to generate potential outliers to help a binary classifier in accurately detect outliers from normal data [14]. Recent works combine graph-based representations with deep learning techniques [15] and fuzzy clustering methods optimized by Particle Swarm Optimization [16] to detect anomaly behavior on time series.

There are some real use cases in which there are not enough data for learning end-to-end methods. Although domain adaptation and transfer learning techniques can be used to deal with small datasets, the success of these end-to-end methods relies on their capability to learn specific features for the outlier detection task. However, there are some use cases, like clinical decision support systems or personalized models, in which feature vector are defined beforehand for the particular task and hence cannot be modified to detect outliers.

2.2. Local methods

Local methods are based on a description of each sample's neighborhood usually defined using the Euclidean distance among samples in the feature space. Given that local methods define outliers in terms of such distances, they are distribution free and, thus, better suited for unbalanced small datasets. The DBSCAN method is a distance-based method able to discover clusters of arbitrary shape while detecting outliers [17].

Most local approaches, like [18] or [19], define outliers in terms of the distance to the k-th nearest point. LOF is an outlier detector method defined in the context of knowledge discovery in databases that assigns an outlierness score to each sample based on local information [20]. This score is computed in terms of the distance to the k nearest neighbors of each point, meaning a score near to 1 to not be an outlier while higher values provide higher certainty of being it. LOCI also bases on k-th nearest neighbors to define a multi-granularity deviation factor (MDEF) as outlierness measure [21]. The MDEF is the relative deviation of sample's local neighborhood density from the average local neighborhood density, so that a point is an outlier if its MDEF is sufficient large. This way LOCI is effective to detect point outliers and collective outliers, as well. The Isolation Forest, IF, technique builds a tree that isolates *attribute* outliers using a binary search [22]. Since an attribute outlier has different values compared to inlier points IF detects them as points such that the length path to reach them is significantly shorter than the mean length to reach any other point.

The selection of the parameters defining neighborhoods is a main bottleneck in local approaches. In particular, the selection of the number k of nearest neighbors is crucial since it greatly affects methods performance. Therefore, several strategies for optimal selection of the parameter k have been proposed since the early years of nearest neighborhood approaches.

2.3. Graph based methods

Neighboring relationship can be defined in terms of distances but also in terms of *friendship* relationship on structured data, like graphs. In structured data, outliers are also linked to topological variation of subgraphs that broke a repetitive pattern [46]. To detect these local structural singularities, it has also been proposed outlier detectors for graph-based representations. Deep learning has also been applied on such representations for anomaly detection [23].

The early work in [24] studied the relationship between connectivity of mutual-knn (mkNN) graph and outlier detection, providing a criteria in terms of the graph topology. In particular, they studied the geometric properties of the underlying data points distribution and derived a theoretical criterium to set a value of k ensuring that the connected components of the graph correspond to clusters in the feature space. In that context, outliers were those samples which did not belong to any of this connected components, that is, they correspond to single node connected components. The work in [24] was later extended in [25]. There, the authors provide further insights on the mutual k-nn graphs to derive tighter bounds to estimate the optimal k to build a mkNN. A main inconvenience for a practical use is that these bounds are still hard to compute with real data having class outliers.

In [26], the authors propose an algorithm to search the optimal k to build the mkNN graph. In that work, the authors introduce the concepts of *stability state* and *appropriate k (apk)* for a mkNN graph and they propose an algorithm to search the optimal k. Finally, the most recent work of [27] also proposes a variation of mkNN graph to minimize the impact of k. In their approach, they compute multiple local proximity graphs for k sampled uniformly in a range of values. Then, their approach does not rely in finding the *optimal k* of the mkNN graph but in combining the information of all mkNN graphs using a random walk to detect outliers.

Aside fixing the parameter k, another concern about existing local approaches is that outlier scores are defined from the structure of a single neighborhood defined using distances. From a mathematical point of view, this implies that the topological structure of feature spaces is modeled as a norm or metric space [28]. Although Euclidean spaces admit a topology defined from a metric or norm, these approaches might fail to properly describe more complex spaces (like manifolds [28]). Thus, topology is a powerful mathematical approach to model the structure of complex manifolds without the assumption of any parametric model for the data.

Methods for the detection of communities in social networks can provide a mean to extract a set of topological neighbors from mkNN graphs. In this context, attribute outliers are often non-connected, or hardly connected, individuals. Meanwhile, class outliers correspond to community members with a user profile, or interests, far of most of community members and they are often ignored. While detecting attribute outliers is done using graph topological properties, computation of a topology in non-structured spaces given by a discrete set of population samples still remains a challenge for topology. In [8], the authors presented a local method based on neighborhoods given by the communities of the graph built from the samples distances. Despite the promising results in the diagnosis of lung cancer in confocal images, this topology given by graph communities is prone to exclude many points that could not be considered isolated attribute outliers [29]. Besides, like other local methods, a main concern is the impact of the parameters used to compute the graph used to detect communities. In case k is too small, communities might exclude points that are not actually outliers [29], while increasing k produces a single community including all points. Thus, the method requires a proper accurate value for the parameter k, which the authors fine-tuned to give optimal results.

Direct analysis of the local structural properties of graphs and networks also allows the detection of outliers and anomalies. The OddBall method is a widely used method for anomaly detection in networks which focus on detecting nodes having topological properties significantly different compared to neighboring nodes [30]. More recently, in [31], the authors propose an extension of the NetEMD network method [32] to detect graph anomalies and spectral localization statistics in financial transaction networks. Other methods base on the Minimum Description Length (MDL) principle. In [33] anomalous sub-graphs are detected using variants of MDL. In [34], the authors use MDL as well as other probabilistic measures to detect several types of graph anomalies (e.g. unexpected/missing nodes/edges).

2.4. Multiview methods

All methods described above are specifically designed for the detection of attribute outliers in single view problems. Multimodal, or multiview, data can benefit of the different sources of information to detect data outliers. In particular, class outliers are easily spotted when a sample is labeled differently across views. In this context, in [35] the authors detect class and attribute outliers based on the spectral analysis of the combined adjacency matrix. In that paper, those samples that lie in the kernel space of the combined adjacency matrix are identified as outliers. A different approach is the one proposed in [36] and [5]. In that works, the authors propose a generalized K-means method that learns cluster label consistencies across views. Samples having different cluster labels are classified as class outliers. A specific limitation to multi view methods is the combination of information across views, which usually leads to under-detection of abnormalities arising in single views. Finally, another limitation is that being designed for more than view, multi view methods are prone to perform poorly in single view problems.

3. COD: outlier detection based on a topological measure of pattern diversity

Our method is a local approach based on graph communities, which encode the structure of a feature space. Fig. 1 sketches the main steps of our method for the two-dimensional single view space shown in Fig. 1.a. The dataset has 2 classes (black dots and red crosses), one class outlier (numbered 6) and one attribute outlier (numbered 10). The 3 main steps of the proposed method applied to the synthetic data are shown in Fig. 1.b-d and described in the pseudocode of the algorithm of Algorithm 1. First, we encode the local structure of feature vectors through a graph representing their mutual k-nearest neighborhood. The nodes of the graph are colored using each class color (red and black). Second, we use methods for dynamical analysis of social networks to extend graph communities from an initial set of communities. Fig. 1.c shows the initial set of communities on the left and their extension on the right image. The algorithmic details of the extension of the initial communities are given in Algorithm 2. These extended communities define, in the original feature space, a set of neighborhoods of each sample. By definition of class and attribute outlier, isolated nodes not belonging to any community are attribute outliers, while class outliers should belong to communities with an heterogeneous distribution of labels. In order to characterize the latter, we define a local measure of abnormality from two probabilistic measures of each sample heterogeneity computed in its set of neighborhoods (see Algorithm 3 for a pseudocode of this step). These two measures define a function from the set of samples to the unit square that maps inliers and outliers to different corners of the square. A classifier, C, discriminating between inliers and outliers provides our measure of outlierness, Fig. 1.d.

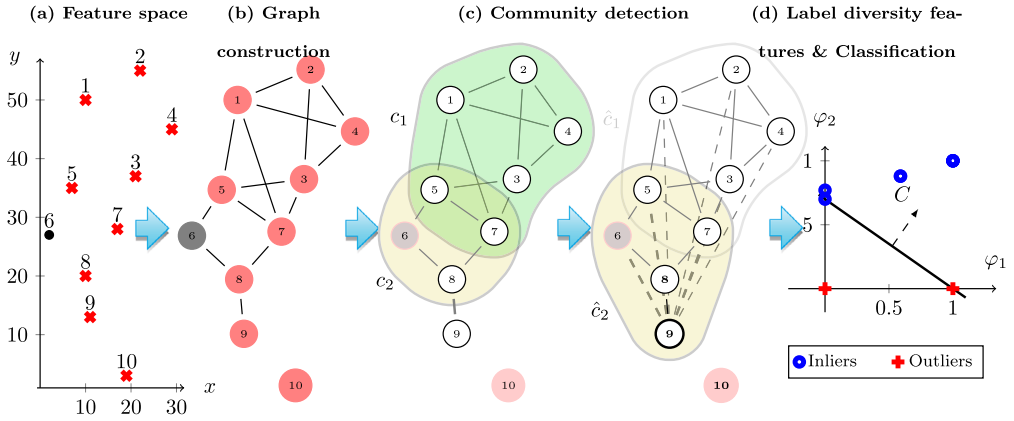


Fig. 1. Overview of the method: (a) Two-dimensional feature space with 2 classes (black dots and red crosses). (b) Graph encoding mutual k-nearest neighbors with nodes colored in red and black according to its class. Nodes 6 and 10 correspond, respectively, to the class attribute and attribute outliers. (c) Detection of communities: initial communities (c_1, c_2) in left graph and their extension ($\hat{c}_1$, $\hat{c}_2$) in the right graph. (d) Feature space and classifier margin (C) giving the final outlierness measure from the communities. (For interpretation of the colors in the figure(s)and table(s), the reader is referred to the web version of this article.)

Algorithm 1: COD: Community-based outlier detector (Single-View)

Data: V: labeled nodes, δ: percentile
Result: s: outlier scores
$A \leftarrow \text{mkNN}(V)$; /* mutual-knn adjacency matrix */
$\mathcal{C} \leftarrow \texttt{CommunityDetection}(A)$;
$\hat{\mathcal{C}} \leftarrow \texttt{ExtendCommunities}(G, \mathcal{C}, \delta)$;
$\Phi \leftarrow \texttt{ComputeTopologicaFeatures}(V, \hat{\mathcal{C}})$;
for $\forall \varphi_i \in \Phi$ **do**
  $s_i \leftarrow \texttt{Classify}(\varphi_i)$;
end

In what follows, we give details about each of the main steps of the proposed algorithm: graph construction, community detection and definition of a outlierness feature space for the classification of inliers and outliers samples.

3.1. Graph construction

The graph, $G = (V, A)$, is given by the adjacency matrix of the mutual k-nearest neighbor of samples. Let $V := \{(\mathbf{v}^i, \ell_{\mathbf{v}^i}) \mid \mathbf{v}^i = (v_1^i, \ldots, v_n^i) \in \mathbb{R}^n,\ \ell_{\mathbf{v}^i} \in \{1, \ldots, L\},\ i = 1, \ldots, N\}$ be a set of N labeled points in an n-dimensional feature space endowed with a distance, namely d, and L the number of labels. For any positive integer, k, let $\text{kNN}(\mathbf{v}^i)$ denote the set of $\mathbf{v}^i$ k-nearest neighbors and $\text{mkNN}(V^i)$ the set of $\mathbf{v}^i$ mutual k-nearest neighbors defined as:

$$\text{mkNN}(\mathbf{v}^i) := \{\mathbf{v}^j \text{ such that } \mathbf{v}^j \in \text{kNN}(\mathbf{v}^i) \text{ and } \mathbf{v}^i \in \text{kNN}(\mathbf{v}^j)\} \quad (1)$$

Then, the adjacency matrix, $A = (a_{ij}) = (a(\mathbf{v}^i, \mathbf{v}^j))$, codifying G is defined as:

$$a(\mathbf{v}^i, \mathbf{v}^j) = \begin{cases} \frac{1}{1+d(\mathbf{v}^i, \mathbf{v}^j)} & \text{if } \mathbf{v}^j \in \text{mkNN}(\mathbf{v}^i) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

for $d(\mathbf{v}^i, \mathbf{v}^j)$ the distance between $\mathbf{v}^j$ and $\mathbf{v}^i$. We note that the graph edges, a_{ij}, are in $[0, 1]$, being close to 0 if mutual neighbors are far from each other, close to 1, otherwise. This way, the number of neighbors for every node, $\mathbf{v}^i$, is related to the sparseness of the point $\mathbf{v}^i$ in the feature space.

3.2. Community detection

To alleviate the impact of k in the computation of (2), communities are computed using criteria for dynamic computation of communities [37–39] to extend an initial set of communities, see Algorithm 2 for algorithmic details of this procedure. The initial communities are given by any community detection algorithms that generates overlapped communities. These detected communities are prone to exclude many points that are not actual attribute outliers [29]. In order to add them to the set of initial communities, we extend them following a modification of the iLCD community detector proposed in [37]. The iLCD method dynamically updates (extend, create or remove) an initial set of communities using the temporal information in which node and edges appear in a network. The criteria for updating communities are given in terms of the community internal connectivity, as well as, the connections between candidate nodes and community nodes. In our case, an isolated node, $\mathbf{w}^j$, is added to an initially detected community, $c \in \mathcal{C}$, if it fulfills that:

$$\text{CS}(c, \mathbf{w}^j) \geq \delta\, \text{IC}(c) \quad (3)$$

for $\delta \in [0, 1]$ being a tolerance parameter, $\text{IC}(c)$ a measure of the community internal connectivity and $\text{CS}(c, \mathbf{w}^j)$ a measure of the connectivity between $\mathbf{w}^j$ and the community c. Both measures are computed from a function of the degree of the community nodes as follows.

Let $\mathcal{S}$ be the set composed of all nodes that belong to any of the initially detected communities and $G^{\mathcal{S}}$ and G^c the subgraphs induced by $\mathcal{S}$ and c, respectively. Then, for $\forall \mathbf{v}^i \in \mathcal{C}$ we can define the following function, $\rho_c(\mathbf{v}^i)$, measuring its belongingness to the community:

$$\rho_c(\mathbf{v}^i) := \frac{\deg^{G^c}(\mathbf{v}^i)}{\deg^{G^{\mathcal{S}}}(\mathbf{v}^i)} \quad (4)$$

being deg the degree function of a node in a graph. The measure of the internal connectivity of c is defined from $\rho_c(\mathbf{v}^i)$ as:

$$\text{IC}(c) := \sum_{\mathbf{v}^i \in c} \rho_c(\mathbf{v}^i) \quad (5)$$

The measure of the connectivity between $\mathbf{w}^j$ and c is also defined from $\rho_c(\mathbf{v}^i)$ as:

$$\text{CS}(c, \mathbf{w}^j) := \sum_{\mathbf{v}^i \in c} \frac{\rho_c(\mathbf{v}^i)}{d(\mathbf{w}^j, \mathbf{v}^i) + 1} \quad (6)$$

since $\text{CS}(c, \mathbf{w}^j)$ is a weighted average of $\rho_c(\mathbf{v}^i)$ with weights $\frac{1}{d(\mathbf{w}^j, \mathbf{v}^i)+1}$, we have that:

$$\mathrm{CS}(c, \mathbf{w}^j) \geq \left(\frac{1}{\max_{ji} d(\mathbf{w}^j, \mathbf{v}^i) + 1} \right) \mathrm{IC}(c) \tag{7}$$

By the above inequality, we could set $\delta = \frac{1}{\max_{ji} d(\mathbf{w}^j, \mathbf{v}^i)+1}$. However, such extreme value could aggregate some attribute outliers to the initial set of communities. In order to avoid such an artifact, we propose to define δ as a percentile of $\frac{1}{d(\mathbf{w}^j, \mathbf{v}^i)+1}$ probabilistic distribution.

Finally, candidate nodes $\mathbf{w}^j$ that satisfy inequality (3) for a given community c are added in the extended community $\hat{c}$. We denote by $\hat{\mathcal{C}}$ the set of all the extended communities.

Algorithm 2: ExtendCommunities

```
Data: G = (V, A): graph , C: set of communities, δ: percentile
Result: Ĉ: set of extended communities
S ← GetNonIsolatedNodes(V, C);
W ← V \ S ;
if W is empty then
    return C ;   /* if no isolated nodes exist, no need to
      extend communities */
end
Ĉ ← ∅ ;
for ∀c ∈ C do
    ĉ ← c ; /* Initialize the extended community with the
     original community */
    ρ_c ← ComputeRho(G, S, v) ;                    /* as in (4) */
    IC(c) ← ComputeIC(c, ρ_c) ;                    /* as in (5) */
    for ∀w^j ∈ W do
        CS(w^j, c) ← ComputeCS(w^j, c, ρ_c) ;      /* as in (6) */
        if (3) holds then
            ĉ ← {w^j} ⋃ ĉ;
        end
    end
    Ĉ ← {ĉ} ⋃ Ĉ;
end
return Ĉ
```

3.3. Outlierness feature space

Given that communities define a set of neighbors in the feature space, nodes not belonging to any of the extended communities correspond to attribute outliers. Meanwhile, class outliers are expected to belong to communities with either high heterogeneity in nodes label or the majority of nodes with a label different from the class outlier label.

Under the above considerations, we define two quantities, φ_1, φ_2, measuring how homogeneous the labels of the communities a sample belongs to are. Algorithm 3 provides the pseudocode describing how to compute them. For each sample, the function φ_1 quantifies the heterogeneity in community labels, while φ_2 quantifies how many nodes in the community have a label different from the sample label. Both measures are based on the probabilistic distribution of the community node labels and are normalized in $[0, 1]$ in such a way that they define a function φ:

$$\begin{array}{rccc} \varphi: & V & \longrightarrow & [0,1] \times [0,1] \\ & (\mathbf{v}^i, \ell_{\mathbf{v}^i}) & \mapsto & (\varphi_1(\mathbf{v}^i, \ell_{\mathbf{v}^i}), \varphi_2(\mathbf{v}^i, \ell_{\mathbf{v}^i})) \end{array} \tag{8}$$

mapping inliers to $(1, 1)$, attribute outliers to $(0, 0)$ and class outliers to either $(1, 0)$ or $(0, 1)$ depending on whether they belong to one or more communities. In case $\mathbf{v}$ does not belong to any community, we have an attribute outlier and, thus, we set $\varphi(\mathbf{v}) = (0, 0)$. This way, φ defines a 2-dimensional feature space able to discriminate inliers and outliers. The probability of a classifier trained to discriminate between them provides our outlierness score and its output class our outlier detection.

The function φ_1 measuring the heterogeneity in community labels is computed from their entropy as follows. For each sample $\mathbf{v}$, let $\mathcal{S}_\mathbf{v} \subseteq \hat{\mathcal{C}}$ denote the subset of communities containing $\mathbf{v}$ and Ent_c the entropy of the labels of the nodes in a community $c \in \mathcal{S}_\mathbf{v}$ defined as:

$$Ent_c = -\sum_{i=1}^{n_l} p_i^c \log(p_i^c) \tag{9}$$

for p_i^c the probability in c of the i-th label. This probability is approximated by the proportion of c nodes that are labeled i:

$$p_i^c := \frac{|\{\mathbf{w} \in c \quad | \quad \ell_\mathbf{w} = i\}|}{|\{\mathbf{w} \in c\}|} \tag{10}$$

for $|\cdot|$ denoting the cardinality of a set. Then, φ_1 is defined as:

$$\varphi_1(\mathbf{v}, \ell_\mathbf{v}) := \varphi_1(\mathbf{v}) := \frac{1}{|\mathcal{S}_\mathbf{v}|} \sum_{c \in \mathcal{S}_\mathbf{v}} \pi(Ent_c \leq T) \tag{11}$$

being $T \in [0, \log(n_\ell)]$ a tolerance threshold on the maximum entropy allowed in community labels and $\pi(a) = 1$ if a holds. We note that $\log(n_\ell$ is the maximum value of $Ent_\mathcal{C}$ achieved in case of uniformly distributed labels, $p_i^c = 1/n_\ell$, $\forall i$.

The score $\varphi_1(\mathbf{v}) \in [0, 1]$ has extreme values $\varphi_1(\mathbf{v}) = 0$ in case all communities in $\mathcal{S}_\mathbf{v}$ have heterogeneous labels and $\varphi_1(\mathbf{v}) = 1$ in case the label within each community is the same for all nodes in the community. We note that, in this last case, by definition of Ent_c, the label must be the same for all communities in $\mathcal{S}_\mathbf{v}$.

The function φ_2 measuring how many nodes in the community have a label equal to the sample label, $\mathbf{v}$, is computed from the probability of $\ell_\mathbf{v}$, as follows. For each $c \in \mathcal{S}_\mathbf{v}$, let $p^c_{\ell_\mathbf{v}}$ denote the probability of $\ell_\mathbf{v}$ in c excluding $\mathbf{v}$:

$$p^c_{\ell_\mathbf{v}} = \frac{|\{\mathbf{w} \in c \setminus \mathbf{v} \quad | \quad \ell_\mathbf{w} = \ell_\mathbf{v}\}|}{|\{\mathbf{w} \in c \setminus \mathbf{v}\}|} \tag{12}$$

Then, φ_2 is defined as:

$$\varphi_2(\mathbf{v}, \ell_\mathbf{v}) := \frac{1}{|\mathcal{S}_\mathbf{v}|} \sum_{c \in \mathcal{S}_\mathbf{v}} p^c_{\ell_\mathbf{v}} \tag{13}$$

The measure $\varphi_2(\mathbf{v}, \ell_\mathbf{v}) \in [0, 1]$ and has extreme values $\varphi_2(\mathbf{v}, \ell_\mathbf{v}) = 0$ in case no community in $\mathcal{S}_\mathbf{v}$ is consistent with $\mathbf{v}$ label, $\varphi_2(\mathbf{v}, \ell_\mathbf{v}) = 1$ in case all communities have nodes with labels equal to $\ell_\mathbf{v}$.

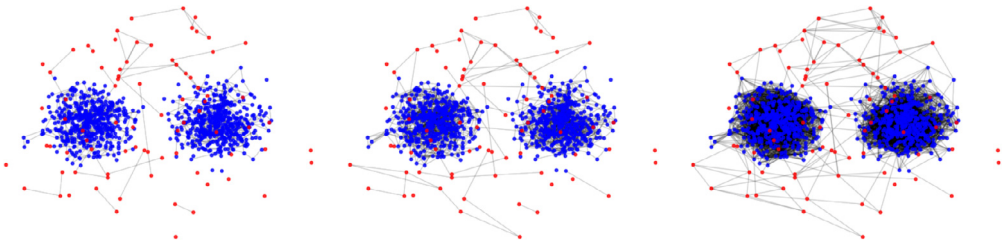
Finally, in order to increase the separability across the different types of outliers, we transform φ_1, φ_2 to logarithmic scale by applying the function:

$$f(x) = \frac{1}{1 - \log x} \tag{14}$$

to $x = \varphi_i$, $i = 1, 2$.

In the multi view case, we model the space as a Cartesian product and, thus, compute a mkNN graph for each view. Communities are computed independently for each of these graphs and so are the probabilistic measures. These view-dependent measures are aggregated to define the function that maps samples to the unit square. We compute the two measures for each view and aggregate them to define φ. If φ_1^j, φ_2^j denote the measures computed for the j-th view, then their minimum values across views define the function φ as:

$$\begin{array}{rccc} \varphi: & V & \longrightarrow & [0,1] \times [0,1] \\ & (\mathbf{v}^i, \ell_{\mathbf{v}^i}) & \mapsto & (\varphi_1(\mathbf{v}^i, \ell_{\mathbf{v}^i}), \varphi_2(\mathbf{v}^i, \ell_{\mathbf{v}^i})) := \\ & & & (\min_j \varphi_1^j(\mathbf{v}^i, \ell_{\mathbf{v}^i}), \min_j \varphi_2^j(\mathbf{v}^i, \ell_{\mathbf{v}^i})) \end{array} \tag{15}$$



(a) $k = 4$ (b) $k = 10$ (c) $k = 40$

Fig. 2. Synthetic data generated for experiments following 2 standard normal distributions. Distribution points have randomly generated and outliers (red points), as well. mkNN graph is generated for values: (a) $k = 4$, (b) $k = 10$ and (c) $k = 40$.

Algorithm 3: ComputeTopologicaFeatures

Data: V: labeled nodes, $\hat{\mathcal{C}}$: set of extended communities
Result: Φ: set of topological features
for $\forall c \in \hat{\mathcal{C}}$ **do**
  $Ent_c \leftarrow$ `ComputeEntropy(`c`)` ; `/* as in (9) */`
end
for $\forall \mathbf{v} \in V$ **do**
  $\varphi_1(\mathbf{v}) \leftarrow$ `ComputePhi1(`$\mathbf{v}, \{Ent_c\}$`)` ; `/* as in (11) */`
  $\varphi_2(\mathbf{v}) \leftarrow$ `ComputePhi2(`$\mathbf{v}, \{Ent_c\}$`)` ; `/* as in (13) */`
end
$\Phi \leftarrow$ `Transform(`Φ`)` ; `/* as in (14) */`
return Φ

4. Experimental set-up

We carried out two experiments to evaluate the performance of the COD method:

1. **Impact of COD configuration in communities.** In a first experiment, we evaluated the benefits of the community extension technique within the community detection step on a synthetic dataset. The performance of COD strongly depend on the detected communities, which in turn also depend on the nodes connectivity. The goal of this experiment was to evaluate the impact of the number of neighbors in the construction of the mkNN graph, as well as, the method for the definition of the initial communities. For this experiment a single view synthetic dataset with attribute outliers was generated.
2. **Comparison to SoA single and multi view methods.** In a second experiment, we compared the COD method to other outlier detection methods on benchmark datasets from the UCI repository.[1] The goal of this experiment was to compare the performance of the proposed method to detect the rate of attribute and class outliers in, both, single and multi view settings. For this experiment a selection of UCI datasets was altered to include several percetages of attribute and class outliers in a single and multi view setting.

Next, we describe the generation of datasets and metrics used for each experiment

4.1. Generation of datasets with outliers

For each experiment, we generated attribute and class outliers following the same experimental setting as in [5]. In order to simulate attribute outliers, the features of the selected samples were changed by random numbers following a distribution with the highest probability outside the range of the values expected for the original data. In the multi view case, features were altered in each view. In order to simulate class outliers, we swapped the labels of points randomly selected from random pairs of classes. In the multi view case, classes were swapped in views randomly selected.

For the first experiment, we generated a single view synthetic dataset in $\mathbb{R}^6$ generated from two standard normal distributions with minimal overlap, see Fig. 2. A total number of 30 different populations of 500 samples for each distribution were generated with different percentages of attribute outliers: 2%, 5% and 8%. Table 1 summarizes the main characteristics of the synthetic dataset used for the first experiment.

For the second experiment, we have considered, both, attribute and class outliers in single and multi view settings. In particular, we considered the following 3 combinations of attribute and class outliers: i) 8% class outlier + 2% attribute outlier, labeled 8-2, ii) 5% class outlier + 5% attribute outlier, labeled 5-2 and iii) 2% class outlier + 8% attribute outlier, labeled 2-8. For the single view setting, we also added the same 3 percentages of attribute outliers as for the synthetic data: 2%, 5% and 8%. Following [35], the multi view case was defined by splitting UCI features into disjoint sets, each of them defining one view. Feature splitting was done by taking half of the original features for each view as reported in [35]. We considered 2 and 3 views for all sets, with the exception of Iris, which dimensionality only allows splitting features in 2-views. For each outlier configuration and UCI dataset, we generated 50 different sets with outliers for statistical analysis of results.

Table 1 reports the main characteristics of the 4 UCI datasets selected for these experiments. These datasets have been selected to be representative of the main configurations of classification feature spaces. In particular, they include datasets presenting the most common artifacts dropping performance of methods, like small sample size (Ionosphere), large dimensionality (Ionosphere) and large number of classes with some of them being minority groups (Letter Recognition).

Table 1
Selected UCI and Synthetic Datasets.

Datasets	Dimension	Num of Samples	Num of Classes
Iris	4	150	3
BCW	10	699	2
Ionosphere	34	351	2
Letter recognition	16	20,000	26
Synthetic	6	1,000	2

4.2. Quality metrics

The metrics used for the first experiment are the false positive (FP) and true positive (TP) rates before and after applying the community extension technique. The FP rate is computed as the ratio of inliers not in a community with respect to all the samples without community. The TP rate is computed as the number

[1] https://archive.ics.uci.edu/ml/datasets.php.

Table 2
Mean and standard deviation of FP and TP rates for the selected CD algorithms before and after applying the community extension step.

CD	k	FP rate			TP rate		
		before	after	reduction rate	before	after	reduction rate
angel [40]	4	0,94 ± 0,03	0,01 ± 0	0,99 ± 0	0,99 ± 0,03	0,98 ± 0,03	0,01 ± 0,02
	10	0,82 ± 0,07	0,01 ± 0,01	0,99 ± 0,01	0,95 ± 0,08	0,86 ± 0,22	0,09 ± 0,22
	40	0,24 ± 0,1	0,23 ± 0,1	0,01 ± 0,03	0,81 ± 0,18	0,81 ± 0,18	0 ± 0
core expansion [41]	4	0,83 ± 0,05	0 ± 0	1 ± 0	0,59 ± 0,19	0,53 ± 0,19	0,11 ± 0,08
	10	0,84 ± 0,04	0 ± 0	1 ± 0	0,48 ± 0,2	0,44 ± 0,2	0,09 ± 0,08
	40	0,55 ± 0,09	0,04 ± 0,04	0,92 ± 0,07	0,32 ± 0,18	0,31 ± 0,18	0,03 ± 0,05
danmf [42]	4	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
	10	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
	40	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
dcs	4	0,89 ± 0,04	0 ± 0	1 ± 0	0,58 ± 0,19	0,53 ± 0,18	0,08 ± 0,06
	10	0,81 ± 0,05	0,02 ± 0,02	0,97 ± 0,02	0,47 ± 0,18	0,46 ± 0,18	0,04 ± 0,05
	40	0,13 ± 0,1	0,13 ± 0,1	0,01 ± 0,04	0,35 ± 0,18	0,35 ± 0,18	0 ± 0,01
k clique	4	0,93 ± 0,03	0 ± 0	1 ± 0	0,94 ± 0,06	0,91 ± 0,07	0,04 ± 0,04
	10	0,78 ± 0,08	0,01 ± 0,01	0,99 ± 0,01	0,88 ± 0,1	0,84 ± 0,11	0,05 ± 0,04
	40	0,19 ± 0,09	0,18 ± 0,09	0,03 ± 0,1	0,69 ± 0,19	0,69 ± 0,19	0 ± 0,01

of outliers without any community with respect the total number of outliers. Thus, the targeted FP rate is 0, which means all inliers points are within a community, and the targeted TP rate is 1, which means that all attribute outliers are not in any community. In other words, setting outliers as the positive class, this metrics measure the percentage of outliers correctly detected by the method and the percentage (over all normal samples) of non-outliers wrongly identified as outliers. Ranges (given by mean ± standard deviation) were computed, for each configuration and community detection approach, over the TP and FP for the 30 populations of synthetic data randomly generated.

The metric used for the second experiment reported in Tables 2-3 is the Area Under the ROC Curve (AUC) computed taking outliers as the positive class. As before, for each method, ranges (given by mean ± standard deviation) were computed over the 50 populations of each outlier configuration: single view with attribute outliers for and multi view with, both, attribute and class outliers.

5. Results

5.1. Experiment 1. Impact of COD configuration in communities

The mkNN graphs were built with values $k = 4$, 10 and 40 on synthetic sets randomly generated as described in Section 4.1. For each graph, we computed the initial communities using 5 off-the-self community detection (CD) algorithms implemented in the CD pyhton Library.[2] For the extension of the initial communities, the tolerance δ in (3) was computed for the 75% percentile. Table 2 reports a statistical summary of TP and FP rates for each COD configuration before and after applying the community extension technique o assess changes in FP and TP rates before and after applying the community extension.

First, given the standard deviation values we observe that FP and TP rates are quite stable through the repetitions and the outliers configuration. Second, the proposed community extension technique behaves as expected, i.e. overall, the FP ratio is reduced without significantly harming the TP ratio regardless the CD algorithm and the value of the mkNN graph. The only exception is the deep learning approach *danmf* which assigns all samples to a single initial community and, thus, the extension technique is useless. This is also the case for $k = 40$ with initial communities defined with the *angel*, *dcs* and *k clique* methods, due to a high internal connectivity of the initial communities. In this case, the $IC(\mathcal{C})$ of the condition to add nodes is close to one and, thus, the inequality given by (5) is hardly satisfied. For lower values of k, FP rate reduces to almost 0 cases, while a reduction in TP rates of less than 10%.

Therefore, the configuration for computing the initial communities that we recommend is a low value of k and any CD giving a large number of initial communities.

5.2. Experiment 2. Comparison to SoA single and multi view methods

We have compared COD to single view methods including 5 local and 3 global approaches. The local methods are DBSCAN [17], LOF [20], LOCI [21], KNN [18] and IF [22], while the global ones are GMM [9], the graph-based APS [27] and the deep-learning one SO-GAAL [14] model. The multi view methods are the best performers reported in [5], DMOD [5], AP [43] and MLRA [44], and the pioneering multi view approach HOAD [35]. According to the first experiment, our method was computed using the following parameters. The mkNN graph was computed setting $k = 10$. For the extension of the initial communities, the tolerance δ in (3) was computed for the 75% percentile as before. Initial communities were computed using the k-clique method.

Finally, for the computation of the outlierness measure, we use a SVM classifier trained on a sub-set of the MirFlickr dataset [45] altered to have the different types of outliers. To train it, we use the second last layer of a pre-trained VGG network as feature vector on images with a single annotation. Then, we applied a K-means to identify those annotations that does not highly overlap between them and, to further reduce the size of samples, we select only those samples with minimal overlapping. Then, we generated class and attribute outliers as explained in Section 4.1.

Table 3 reports the AUC (mean ± standard deviation) for the results obtained for the proposed COD, DBSCAN, LOF, LOCI, IF, GMM, SO-GAAL and APS on Iris, Breast Cancer Winconsin (BCW), Ionosphere and Letter Recognition datasets with different settings [27]. Following [5], the first and second best performers for each configuration are marked in red and blue, respectively. The analysis of the ranges lead to the following observations.

According to the number of times that methods are ranked in the top 2, best performers are COD (being $12/24 = 50\%$ times with $10/24 = 42\%$ first rank), KNN (being $13/24 = 54\%$ times with $3/24 = 12\%$ first rank), GMM (being $7/24 = 30\%$ times with $3/24 = 12\%$ first rank), APS (being $5/24 = 20\%$ times with $4/24 = 17\%$

[2] CDlib - Community Discovery Library: https://cdlib.readthedocs.io/.

Table 3
AUC values (mean ± standard deviation) for the Single View Case.

DataSet	Method	2-8	5-5	8-2	0-8	0-5	0-2
Iris	DBSCAN	0.716 ± 0.074	0.675 ± 0.050	0.580 ± 0.027	0.805 ± 0.103	0.831 ± 0.100	0.883 ± 0.130
	LOF	0.973 ± 0.020	0.958 ± 0.024	0.949 ± 0.039	0.992 ± 0.003	0.978 ± 0.003	0.958 ± 0.024
	LOCI	0.962 ± 0.022	0.888 ± 0.052	0.728 ± 0.058	0.971 ± 0.007	0.966 ± 0.011	0.962 ± 0.006
	KNN	0.974 ± 0.019	0.936 ± 0.042	0.866 ± 0.061	0.990 ± 0.002	0.982 ± 0.004	0.970 ± 0.002
	IF	0.905 ± 0.025	0.855 ± 0.029	0.814 ± 0.036	0.987 ± 0.009	0.975 ± 0.001	0.959 ± 0.000
	APS	0.882 ± 0.047	0.891 ± 0.046	0.882 ± 0.047	0.862 ± 0.07	0.854 ± 0.088	0.910 ± 0.120
	GMM	0.484 ± 0.004	0.484 ± 0.004	0.485 ± 0.012	0.484 ± 0.003	0.484 ± 0.003	0.484 ± 0.002
	SO-GAAL	0.614 ± 0.019	0.605 ± 0.096	0.559 ± 0.087	0.663 ± 0.090	0.631 ± 0.090	0.602 ± 0.134
	COD	0.976 ± 0.045	0.980 ± 0.029	0.979 ± 0.020	0.971 ± 0.052	0.980 ± 0.041	0.989 ± 0.007
BCW	DBSCAN	0.845 ± 0.025	0.724 ± 0.016	0.590 ± 0.008	0.936 ± 0.030	0.944 ± 0.034	0.970 ± 0.032
	LOF	0.545 ± 0.092	0.528 ± 0.071	0.509 ± 0.032	0.513 ± 0.071	0.525 ± 0.088	0.511 ± 0.044
	LOCI	0.882 ± 0.008	0.735 ± 0.010	0.593 ± 0.013	0.981 ± 0.005	0.972 ± 0.004	0.963 ± 0.002
	KNN	0.889 ± 0.005	0.739 ± 0.008	0.592 ± 0.012	0.991 ± 0.002	0.981 ± 0.002	0.966 ± 0.001
	IF	0.885 ± 0.005	0.733 ± 0.009	0.588 ± 0.013	0.987 ± 0.001	0.973 ± 0.001	0.959 ± 0.000
	APS	0.794 ± 0.029	0.674 ± 0.029	0.580 ± 0.029	0.882 ± 0.033	0.863 ± 0.040	0.895 ± 0.048
	GMM	0.371 ± 0.008	0.472 ± 0.015	0.574 ± 0.015	0.299 ± 0.003	0.299 ± 0.002	0.298 ± 0.002
	SO-GAAL	0.693 ± 0.040	0.597 ± 0.032	0.533 ± 0.023	0.733 ± 0.056	0.709 ± 0.063	0.672 ± 0.084
	COD	0.879 ± 0.039	0.830 ± 0.035	0.747 ± 0.033	0.936 ± 0.042	0.954 ± 0.037	0.978 ± 0.002
Ionosphere	DBSCAN	0.850 ± 0.012	0.721 ± 0.015	0.582 ± 0.017	0.954 ± 0.007	0.952 ± 0.007	0.947 ± 0.010
	LOF	0.679 ± 0.034	0.647 ± 0.030	0.572 ± 0.019	0.736 ± 0.041	0.797 ± 0.047	0.892 ± 0.059
	LOCI	0.823 ± 0.025	0.722 ± 0.016	0.585 ± 0.020	0.922 ± 0.027	0.956 ± 0.023	0.954 ± 0.018
	KNN	0.874 ± 0.009	0.734 ± 0.011	0.583 ± 0.017	0.987 ± 0.005	0.976 ± 0.002	0.960 ± 0.001
	IF	0.871 ± 0.011	0.734 ± 0.012	0.584 ± 0.019	0.987 ± 0.006	0.974 ± 0.001	0.958 ± 0.010
	APS	0.881 ± 0.025	0.742 ± 0.026	0.579 ± 0.034	0.995 ± 0.002	0.996 ± 0.004	0.996 ± 0.005
	GMM	0.590 ± 0.012	0.697 ± 0.017	0.814 ± 0.019	0.497 ± 0.001	0.497 ± 0.001	0.497 ± 0.001
	SO-GAAL	0.515 ± 0.035	0.503 ± 0.029	0.511 ± 0.025	0.514 ± 0.048	0.509 ± 0.043	0.503 ± 0.055
	COD	0.845 ± 0.023	0.762 ± 0.032	0.686 ± 0.033	0.902 ± 0.007	0.891 ± 0.006	0.890 ± 0.003
Letter Rec.	DBSCAN	0.891 ± 0.005	0.744 ± 0.003	0.606 ± 0.0018	0.994 ± 0.005	0.995 ± 0.0059	0.997 ± 0.008
	LOF	0.533 ± 0.009	0.522 ± 0.007	0.508 ± 0.007	0.539 ± 0.011	0.527 ± 0.013	0.504 ± 0.015
	LOCI	±	±	±	±	±	±
	KNN	0.513 ± 0.010	0.500 ± 0.008	0.486 ± 0.006	0.516 ± 0.012	0.510 ± 0.012	0.521 ± 0.014
	IF	0.516 ± 0.013	0.503 ± 0.010	0.494 ± 0.009	0.514 ± 0.013	0.509 ± 0.016	0.534 ± 0.019
	APS	0.492 ± 0.014	0.483 ± 0.010	0.473 ± 0.011	0.503 ± 0.015	0.640 ± 0.015	0.751 ± 0.014
	GMM	0.894 ± 0.010	0.893 ± 0.009	0.901 ± 0.008	0.916 ± 0.011	0.939 ± 0.011	0.968 ± 0.012
	SO-GAAL	0.495 ± 0.021	0.492 ± 0.020	0.487 ± 0.018	0.485 ± 0.024	0.493 ± 0.035	0.492 ± 0.043
	COD	0.844 ± 0.013	0.872 ± 0.009	0.906 ± 0.007	0.821 ± 0.017	0.809 ± 0.015	0.795 ± 0.013

first rank) and DBSCAN (being $5/24 = 20\%$ times with $3/24 = 12\%$ first rank). The remaining methods achieve top ranges in less than 20% of the cases, being the deep-learning method SO-GAAL the worst performer with 0 top ranges. The ranking of top ranges indicates that local methods perform better than global ones, as 3 best ranked methods are local approaches. It is worth noticing that our COD is the only method that has top ranges for all datasets and outlier configurations. Meanwhile, the other top ranked methods achieve their best performance only in some of the datasets. In particular, LOCI did not converge for the Letter Recognition dataset.

If we compare ranges between configurations with same quantity of attribute outliers but different quantity of class outliers (2-8 against 0-8, 5-5 against 0-5 and 8-2 against 0-2) we have that, in general, community detection methods are better detecting attribute outliers than class outliers. In particular, the configuration that has worst ranges is the one with higher number of class-outliers (8-2). Among all methods, our COD is the one that has the lowest drop in performance in the presence of class-outliers.

Table 4 reports the AUC (average ± standard deviation) for the results obtained for the proposed COD, HOAD, AP, MLRA and DMOD on Iris, Breast Cancer Winconsin (BCW), Ionosphere and Letter Recognition with different settings. As before, the best two performers for each configuration are marked in red and blue. The analysis of the ranges lead to the following observations.

Ranking as before according to the number of top ranges, the best performers by large are COD ($21/21 = 100\%$ times with $18/21 = 86\%$ first ranks) and DMOD ($17/21 = 81\%$ times with only $2/21 = 10\%$ first ranks). The proposed COD is the best performer for all cases, but three cases (2-View Ionosphere with outlier configuration 2-8 and 3-View Letter Recognition with outlier configuration 8-2) that it is the second best. The HOAD method is the worst performer with 0 times having top ranges.

All methods, excluding the proposed COD, perform better in the 2-view case. Our COD is the only method that has similar (top) ranges for, both, 2 and 3 view configurations. Regarding outlier configurations, there is not a clear trend across datasets. For Iris and Breast performance increases with the number of class outliers, while it decreases for Ionosphere and Letter Recognition datasets. Although this behavior holds for both views, the decrease rate seems to be a bit higher for the 3-view case. Therefore, we attribute it to, both, the separability of the original dataset, as well as, the partition of features to simulate the multi view configuration, which might have selected features having the least discriminating power.

6. Discussion and conclusions

The first experiment shows that our community extension eliminates FP in the detection of attribute outliers in case of initial communities with low connectivity. In such case, the rate of TP for the initial communities is already very high and close to one. Low connectivity in communities is guaranteed for low values of k in mkNN graph, regardless of the DC method used. Therefore, since the community extension preserves TP rates, we conclude that COD has optimal performance for $k \leq 10$ regardless of the DC method used for the computation of initial communities. Consequently, this makes the choice of community detection algorithm and the values of the mkNN graph to be not critical for the final performance of the COD method.

Table 4
AUC values (mean ± standard deviation) for the multi view Case.

DataSet	Method	2-View Case			3-View Case		
		2-8	5-5	8-2	2-8	5-5	8-2
Iris	HOAD	0.167 ± 0.057	0.309 ± 0.063	0.430 ± 0.055	−−	−−	−−
	AP	0.326 ± 0.027	0.630 ± 0.021	0.840 ±0.021	−−	−−	−−
	MLRA	0.856 ± 0.063	0.828 ± 0.080	0.826 ± 0.089	−−	−−	−−
	DMOD	0.909 ± 0.044	0.831 ± 0.038	0.799 ± 0.068	−−	−−	−−
	COD	0.975 ± 0.024	0.971 ± 0.023	0.970 ± 0.021	−−	−−	−−
BCW	HOAD	0.555 ± 0.072	0.586 ± 0.061	0.634 ± 0.046	0.538 ±0.027	0.597 ± 0.038	0.643 ± 0.008
	AP	0.293 ± 0.012	0.532 ± 0.024	0.693 ± 0.023	0.190 ± 0.016	0.388 ± 0.012	0.593 ± 0.046
	MLRA	0.745 ± 0.056	0.715 ± 0.022	0.688 ± 0.028	0.614 ± 0.057	0.596 ± 0.032	0.599 ± 0.029
	DMOD	0.824 ± 0.022	0.752 ± 0.019	0.692 ± 0.036	0.657 ± 0.017	0.720 ± 0.013	0.799 ± 0.016
	COD	0.890 ± 0.027	0.935 ± 0.019	0.947 ± 0.013	0.838 ± 0.022	0.897 ± 0.020	0.910 ± 0.014
Ionosphere	HOAD	0.446 ± 0.074	0.442 ± 0.051	0.448 ± 0.041	0.489 ± 0.079	0.477 ± 0.072	0.444 ± 0.065
	AP	0.623 ± 0.033	0.761 ± 0.025	0.822 ± 0.030	0.511 ± 0.027	0.659 ± 0.043	0.758 ± 0.035
	MLRA	0.645± 0.084	0.669 ± 0.028	0.776± 0.037	0.645 ± 0.040	0.663 ± 0.048	0.700 ± 0.045
	DMOD	0.877 ± 0.032	0.801 ± 0.042	0.774 ± 0.049	0.818 ± 0.018	0.787 ± 0.039	0.784 ± 0.037
	COD	0.841 ± 0.024	0.811 ± 0.024	0.780 ± 0.029	0.854 ± 0.019	0.827 ± 0.025	0.791 ± 0.036
Letter Rec.	HOAD	0.536 ± 0.046	0.663 ± 0.057	0.569 ± 0.049	0.193 ± 0.022	0.488 ± 0.111	0.563 ± 0.081
	AP	0.372 ± 0.057	0.550 ± 0.043	0.640 ± 0.051	0.189 ± 0.039	0.340 ± 0.037	0.570 ± 0.63
	MLRA	0.883 ± 0.024	0.817± 0.051	0.786 ± 0.065	0.841 ± 0.055	0.716 ± 0.044	0.640 ± 0.081
	DMOD	0.912 ± 0.029	0.846 ± 0.022	0.762 ± 0.025	0.916 ± 0.031	0.815 ± 0.038	0.664 ± 0.037
	COD	0.926 ± 0.009	0.904 ± 0.011	0.877 ± 0.011	0.843 ± 0.014	0.816 ± 0.016	0.774 ± 0.017

Concerning comparison to SoA methods in single and multi view settings, the analysis of results lead to the following conclusions. In, both, single and multi view settings, local methods perform better than global ones. In fact, the worst performers in both settings are the global methods SO-GAAL (single view) and HOAD (multi view). Among all local approaches, COD outperforms existing methods in, both, single and multi view settings, regardless of the outlier configuration. Unlike most methods, it performs equally well in small size and high dimensionality datasets (like Ionosphere). COD is also top performer in case of minority classes (like the Letter Recognition dataset) for multi view settings. Although for single view configurations performance drops, COD is still competitive compared to top performers.

It is worth noticing that COD has been applied with the same parameter configuration to all datasets in, both, training and test. Regarding COD training, we used a completely different repository (MirFlickr-25K dataset) and no learning transfer was applied. This is a main advantage compared to existing approaches that require fine tuning of parameters and a re-training for new datasets.

All in all, comparing to existing outliers detection methods, the proposed approach has the following advantages. On one side, our description of feature spaces local structure endows COD with the following properties:

- Capability to model data distributions structured as manifolds. Our description based on mkNN graph communities provides, for each sample, with a set of multiple neighborhoods alternative to the usual n-dimensional Euclidean balls centered at each sample. COD alternative neighborhoods can describe data distributions structured as manifolds, which is a significant advantage over existing local methods.
- High performance in small sample size unbalanced sets. Unlike global and deep learning methods, graph communities can be robustly computed for small data sets regardless of the dimension of feature spaces.

On the other side, the normalized representation of outlierness has the following significance:

- Ability to detect class, as well as, attribute outliers in single and multi view settings. With the exception of [5], existing methods are specialized in detecting a unique type of outliers. The only method [5] able to detect both types only works for multi view settings. COD representation space measuring diversity from topological neighborhoods allows the detection of all kinds of outliers regardless of the view structure.
- A universal outlierness measure that can be directly applied to any data set without any exta training or fine tuning. Since COD is based on the distribution of intrinsic measures of heterogeneity, the only adaptation that requires is the computation of mkNN graph communities of each new data set. This is a main advantage over most existing methods that require training the model from scratch for each new data set.

In summary, we propose an outlier detector based on a topological description of local structure to provide a universal outlierness representation space able to detect all types of outliers even in small size unbalanced settings which are specially challenging for most machine learning methods.

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.

Acknowledgement

This work was supported by the Spanish Ministry of Science and Innovation with projects RTI2018-095645-B-C21, RTI2018-095209-B-C21, PID2021-126776OB-C21 and PID2021-12688OB-I00, Generalitat de Catalunya, 2017-SGR-1624 and 2017-SGR-1783 and CERCA-Programme. DGil would like to dedicate this work to the most outstanding outlier, her mother, Esther Resina Enfedaque.

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. 41 (3) (2009) 15.
[2] M. Kopp, M. Grill, J. Kohout, Community-based anomaly detection, in: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2018, pp. 1–6.
[3] R. Francisquini, A.C. Lorena, M.C. Nascimento, Community-based anomaly detection using spectral graph filtering, Appl. Soft Comput. 118 (2022) 108489, https://doi.org/10.1016/j.asoc.2022.108489, https://www.sciencedirect.com/science/article/pii/S1568494622000424.

[4] Z. Chen, W. Hendrix, N.F. Samatova, Community-based anomaly detection in evolutionary networks, J. Intell. Inf. Syst. 39 (1) (2012) 59–85.
[5] H. Zhao, et al., Consensus regularized multi-view outlier detection, IEEE Trans. Image Process. 27 (1) (2018).
[6] P. Lambin, R. Leijenaar, T. Deist, et al., Radiomics: the bridge between medical imaging and personalized medicine, Nat. Rev. Clin. Oncol. 14 (2017) 749–762.
[7] K. Choi, J. Yi, C. Park, S. Yoon, Deep learning for anomaly detection in time-series data: review, analysis, and guidelines, IEEE Access 9 (2021) 120043–120065.
[8] D. Gil, O. Ramos Terrades, E. Minchole, C. Sanchez, N. Cubero de Frutos, M. Diez-Ferrer, A. Maria Ortiz, R. ans Rosell, Classification of confocal endomicroscopy patterns for diagnosis of lung cancer, in: CLIP-MICCAI, in: Lecture Notes in Computer Science, 2017, pp. 151–159.
[9] X. Yang, L. Jan, L.D. Pokrajac, Outlier detection with globally optimal exemplar-based gmm, in: SIAM International Conference on Data Mining, 2009, pp. 145–154.
[10] S. Pandhre, M. Gupta, V.N. Balasubramanian, Community-based outlier detection for edge-attributed graphs, CoRR, arXiv:1612.09435 [abs], 2016, arXiv: 1612.09435, http://arxiv.org/abs/1612.09435.
[11] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: a survey, CoRR, arXiv:1901.0340, 2019.
[12] R. Chalapathy, A. Krishna Menon, S. Chawla, Anomaly detection using one-class neural networks, CoRR, arXiv:1802.06360, 2018.
[13] L. Ruff, N. Görnitz, L. Deecke, S.A. Siddiqui, R. Vandermeulen, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: International Conference on Machine Learning, 2018, pp. 4390–4399.
[14] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, X. He, Generative adversarial active learning for unsupervised outlier detection, CoRR, arXiv:1809.10816, 2018.
[15] A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series, in: Proceedings of the AAAI Conference on Artificial Intelligence 35 (5) (2021) 4027–4035, https://ojs.aaai.org/index.php/AAAI/article/view/16523.
[16] J. Li, H. Izakian, W. Pedrycz, I. Jamal, Clustering-based anomaly detection in multivariate time series data, Appl. Soft Comput. 100 (2021) 106919, https://doi.org/10.1016/j.asoc.2020.106919, https://www.sciencedirect.com/science/article/pii/S1568494620308577.
[17] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, AAAI Press, 1996, pp. 226–231.
[18] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, SIGMOD Rec. 29 (2) (2000) 427–438.
[19] F. Angiulli, C. Pizzuti, Fast outlier detection in high dimensional spaces, in: T. Elomaa, H. Mannila, H. Toivonen (Eds.), Principles of Data Mining and Knowledge Discovery, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 15–27.
[20] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, Lof: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, ACM, New York, NY, USA, 2000, pp. 93–104.
[21] S. Papadimitriou, H. Kitagawa, P.B. Gibbons, C. Faloutsos, LOCI: fast outlier detection using the local correlation integral, in: Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405), 2003, pp. 315–326.
[22] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data 6 (1) (mar 2012).
[23] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q.Z. Sheng, H. Xiong, L. Akoglu, A comprehensive survey on graph anomaly detection with deep learning, IEEE Trans. Knowl. Data Eng. (2021) 1, https://doi.org/10.1109/TKDE.2021.3118815.
[24] M.R. Brito, E.L. Chávez, A.J. Quiroz, J.E. Yukich, Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection, Stat. Probab. Lett. 35 (1) (1997) 33–42.
[25] M. Maier, M. Hein, U. von Luxburg, Cluster identification in nearest-neighbor graphs, in: M. Hutter, R.A. Servedio, E. Takimoto (Eds.), Algorithmic Learning Theory, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 196–210.
[26] J. Ning, L. Chen, C. Zhou, Y. Wen, Parameter k search strategy in outlier detection, Pattern Recognit. Lett. 112 (2018) 56–62.
[27] C. Wang, Z. Liu, H. Gao, Y. Fu, Applying anomaly pattern score for outlier detection, IEEE Access 17 (2019) 16008–16021.
[28] J. Munkres, General Topology, Prentice-Hall, 2000.
[29] J. Mielgo, Analysis of Community Detection Algorithms for Image Annotation, 2017.
[30] L. Akoglu, M. McGlohon, C. Faloutsos oddball, Spotting anomalies in weighted graphs, in: M.J. Zaki, J.X. Yu, B. Ravindran, V. Pudi (Eds.), Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 410–421.
[31] A. Elliott, M. Cucuringu, M.M. Luaces, P. Reidy, G. Reinert, Anomaly detection in networks with application to financial transaction networks, CoRR, arXiv: 1901.00402, 2019.
[32] A. Wegner Ospina-Forero, R. Gaunt, C. Deane, G. Reinert, Identifying networks with common organizational principles, J. Complex Netw. 6 (6) (2018) 887–913.
[33] C.C. Noble, D.J. Cook, Graph-based anomaly detection, in: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03, 2003, pp. 631–636.
[34] W. Eberle, L. Holder, Discovering structural anomalies in graph-based data, in: Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007), 2007, pp. 393–398.
[35] J. Gao, N. Du, W. Fan, D. Turaga, S. Parthasarathy, J. Han, A multi-graph spectral framework for mining multi-source anomalies, in: Graph Embedding for Pattern Analysis, Springer, 2013, pp. 205–228.
[36] 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.
[37] R. Cazabet, F. Amblard, C. Hanachi, Detection of overlapping communities in dynamical social networks, in: IEEE 2nd International Conference on Social Computing, 2010, pp. 309–314.
[38] K.S. Xu, M. Kliger, A.O. Hero, Tracking communities in dynamic social networks, in: J. Salerno, S.J. Yang, D. Nau, S.-K. Chai (Eds.), Social Computing, Behavioral-Cultural Modeling and Prediction, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 219–226.
[39] V. Sekara, A. Stopczynski, S. Lehmann, Fundamental structures of dynamic social networks, Proc. Natl. Acad. Sci. 113 (36) (2016) 9977–9982.
[40] G. Rossetti, Exorcising the demon: angel, efficient node-centric community discovery, in: International Conference on Complex Networks and Their Applications, Springer, 2019.
[41] A. Choumane, A. Awada, A. Harkous, Core expansion: a new community detection algorithm based on neighborhood overlap, Soc. Netw. Anal. Min. 10 (12) (2020), https://doi.org/10.1007/s13278-020-00647-6.
[42] F. Ye, C. Chen, Z. Zheng, Deep autoencoder-like nonnegative matrix factorization for community detection, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 1393–1402.
[43] A.M. Alvarez, M. Yamada, A. Kimura, T. Iwata, Clustering-based in multi-view data, in: ACM Int. Conf. Conf. Inf. Knowl. Manage, 2013.
[44] S. Li, M. Shao, Y. Fu, Multi-view low-rank analysis for outlier detection, in: SIAM Int. Conf. Data Mining, 2015.
[45] M.J. Huiskes, M.S. Lew, The MIR Flickr retrieval evaluation, in: MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval, ACM, 2008.
[46] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3–5) (2010) 75–174
.