An Efficient Representation-Based Method for Boundary Point and Outlier Detection

Xiaojie Li, Jiancheng Lv, *Member, IEEE*, and Zhang Yi, *Fellow, IEEE*

***Abstract*— Detecting boundary points (including outliers) is often more interesting than detecting normal observations, since they represent valid, interesting, and potentially valuable patterns. Since data representation can uncover the intrinsic data structure, we present an efficient representation-based method for detecting such points, which are generally located around the margin of densely distributed data, such as a cluster. For each point, the negative components in its representation generally correspond to the boundary points among its affine combination of points. In the presented method, the reverse unreachability of a point is proposed to evaluate to what degree this observation is a boundary point. The reverse unreachability can be calculated by counting the number of zero and negative components in the representation. The reverse unreachability explicitly takes into account the global data structure and reveals the disconnectivity between a data point and other points. This paper reveals that the reverse unreachability of points with lower density has a higher score. Note that the score of reverse unreachability of an outlier is greater than that of a boundary point. The top-*m* ranked points can thus be identified as outliers. The greater the value of the reverse unreachability, the more likely the point is a boundary point. Compared with related methods, our method better reflects the characteristics of the data, and simultaneously detects outliers and boundary points regardless of their distribution and the dimensionality of the space. Experimental results obtained for a number of synthetic and real-world data sets demonstrate the effectiveness and efficiency of our method.**

***Index Terms*— Boundary points, data representation, disconnectivity, outlier detection.**

I. Introduction

MANY techniques have been developed to assist people in extracting useful information from rapidly growing volumes of digital data [1]–[5]. Unlike the tasks of clustering, classification, and pattern analysis, which aim to find general patterns, the detection of boundary points (including outliers) is generally performed to identify observations that represent valid, interesting, and potentially valuable patterns in data. For example, a liver disorder detection system might consider normal observations as healthy patients, outlier observations as patients with liver disorders, and boundary points as patients who should have developed liver disorders but somehow have not. Such a system would help in the study of the characteristics of the disease, and special attention would be warranted for the set of people corresponding to the outlier and boundary observations.

Outlier detection is an important task in many practical applications, such as fraud detection, public health, and network intrusion [6]. Many definitions of an outlier have been proposed with their seemingly being no universally accepted definition. The classical definition of an outlier was proposed in [7] and states that an outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. Aggarwal and Yu [8] stated that outliers may be considered as noise points lying outside a set of defined clusters or alternatively outliers as the points that lie outside of the set of clusters while also being separated from the noise. Meanwhile, various techniques have been employed with different names, such as novelty detection, anomaly detection, noise detection, deviation detection, and exception mining [9]–[15].

Papers [10]–[12] provided a comprehensive overview of outlier detection approaches. Outlier detection approaches fall roughly into global and local outlier models [10]. The former group makes a binary decision on whether an observation is an outlier, while the latter group first assigns a value to each observation to assess to what degree each observation is an outlier, and then retrieves the top-*m* ranked outliers, which is more interesting and obviously preferable. In this paper, we focus on models in the latter group. As shown in [11], depending on whether labels for outliers are available, outlier detection methods can be classified as supervised, semisupervised, and unsupervised methods. In general, it is more difficult to get a labeled set which can cover all possible type of outlier behavior, and an unsupervised problem is encountered. Alternatively, according to the underlying approach adopted by each technique, existing outlier detection approaches can be divided into six categories [11]: classification models [16]–[18], nearest neighbor models [13], [19], [20], clustering models [21], statistical models [22], [23], information theory models [24], and spectral theory models [25]. Table I summarizes some classes of commonly encountered time complexities [11]. However, some methods cannot be used for high-dimensional data owing to their computational complexity. Others that appear practical for a high-dimensional space rely implicitly or explicitly on distance, and the poor discrimination of distance

Manuscript received July 5, 2015; revised November 1, 2015, February 18, 2016, and July 10, 2016; accepted September 25, 2016. Date of publication October 19, 2016; date of current version December 29, 2017. This work was supported in part by the National Science Foundation of China under Grant 61375065 and Grant 61602066, in part by the State Key Program of National Science Foundation of China under Grant 61432012 and Grant 61432014, and in part by the Scientific Research Foundation of CUIT under Project KYTZ201608. *(Corresponding author: Jiancheng Lv.)*

The authors are with the School of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: lvjiancheng@scu.edu.cn).

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

Digital Object Identifier 10.1109/TNNLS.2016.2614896

TABLE I

COMMON TIME COMPLEXITIES. n DENOTES THE NUMBER OF OBSERVATIONS. d INDICATES THE NUMBER OF DIMENSIONS

Categories	*Computational Complexity*
Classification models	*depends on the classification algorithm*
Nearest neighbor models	O(2^d) *but often* O(n^2)
Clustering models	*depends on the clustering algorithm* *(such as $O(n^2)$ or $O(n)$)*
Statistical models	*depends on the statistical model* *(such as $O(n^2)$ or $O(2^n)$)*
Information theory models	*the basic technique has* O(2^n) *or linear time complexity*
Spectral theory models	O(n) *but often* d^2

in high-dimensional spaces reduces the performance of these methods [6].

In general, boundary points are located near the margin of densely distributed data, such as a cluster. While there is no universally accepted definition of a boundary point, it has been pointed out [26] that boundary points are different from outliers, or their statistical counterpart—the change point. The classical definition of an outlier was proposed in [22] and states that boundary points sit on the extremes of a class region, i.e., near free pattern space. Depending on whether labels for boundary points are available, boundary detection methods can be classified as supervised, semisupervised, and unsupervised methods [22], [27]–[29]. The border-edge pattern selection (BEPS) method, employed in [22], plays an important role in identifying boundary points. Suppose a data set consists of c classes and n points in each class. The time complexity of BEPS is in $O(c \cdot n \cdot k_e)$, where k_e is the number of nearest neighbors used for identifying edge patterns. In general, an unsupervised problem is generally encountered. This paper focuses on unsupervised models. Detecting boundary points is useful in many applications, such as the liver disorder detection system. To the best of our knowledge, some techniques can be used to detect boundary points, but they are applicable in low-dimensional spaces.

However, many areas of pattern recognition, information retrieval, machine learning, and data mining require the analysis of high-dimensional data [6], [30]. Detecting outliers and boundary points in such cases would involve new challenges that are difficult to overcome. A well-adapted solution, applicable to high-dimensional spaces, is to learn a data representation that can uncover the intrinsic data structure. The representation captures sufficient information about the geometry of the high-dimensional data and is fast to approximate and query and easy to interpret [30], [31]. Over the past decade, numerous methods that learn a data representation and perform better than traditional methods in clustering and dimensionality reduction have been established [30], [32]–[38]. Thus, representation-based methods may be useful in overcoming the limitations of traditional detection methods in high-dimensional spaces.

Although outliers and boundary points are different by definition, they are generally located around the margin of the data set with high density, such as a cluster. Since data representation can uncover the intrinsic data structure, and because of the limitations of the traditional detection methods in high-dimensional spaces, this paper proposes an efficient representation-based method that can be used to simultaneously identify both types of points, regardless of their distribution and the dimensionality of the space. For each point, the negative components in its representation generally correspond to the boundary points among the affine combination of points. The method uses a proposed concept of the reverse unreachability of a point to evaluate to what degree the observation is a boundary point. The reverse unreachability is calculated by counting the number of zero and negative components in the representation. The reverse unreachability explicitly takes into account the global data structure and reveals the disconnectivity between a data point and other points. This paper finds that the reverse unreachability of points with low density has a score higher than that of points with high density. Note that the score of reverse unreachability of an outlier is greater than that of boundary points. The top-m ranked points are identified as outliers, which are more interesting and obviously preferable for further investigation. The greater the value of the reverse unreachability, the most likely the point is a boundary point. Compared with related methods, our method better reflects the characteristics of the data and simultaneously detects outliers and boundary points regardless of their distribution and the dimensionality of the space. Experimental results on a number of synthetic and real-world data sets demonstrate the effectiveness and efficiency of our method.

The remainder of this paper is organized as follows. Section II presents preliminaries and the motivation. Our novel boundary detection method is introduced in Section III. In Section IV, the experimental results on a number of synthetic and real-world data sets demonstrate the effectiveness and efficiency of our method. Finally, our conclusions are presented in Section V.

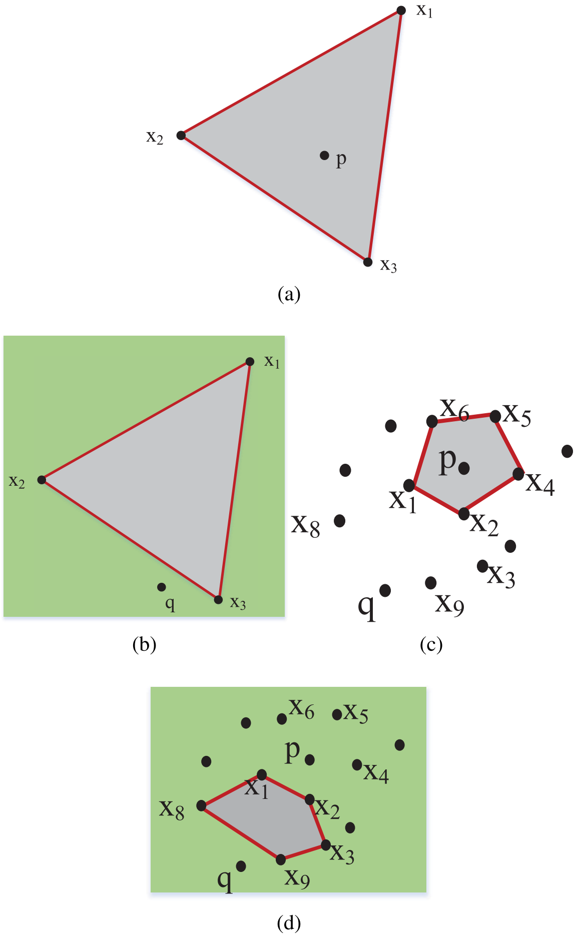
II. PRELIMINARIES AND MOTIVATION

Given data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in R^{d \times n}$, each column vector $\mathbf{x}_i$ denotes an observation, d indicates the number of variables or features, and n indicates the number of observations. For a sample $\mathbf{b}$, the set $U = \{\mathbf{w} \in R^n | \mathbf{Xw} = \mathbf{b}\}$ is an affine space, but not a vector space (linear space) in general [39], where $\mathbf{w}$ is the representation of $\mathbf{b}$. A number of researchers have shown that the negative components in the representation may be helpful for such points (in identifying points) on the boundary of a manifold and outside the convex of their neighbors [40]–[42].

In simple geometric terms, all convex combinations are within the convex hull of the given points [44] [see Fig. 1(a) and (c)]. Formally, the point $\mathbf{p}$ is constructed geometrically as

$$\mathbf{p} = \sum_{i=1}^{n} w_i \mathbf{x}_i, \quad \text{s.t.,} \sum_{i=1}^{n} w_i = 1 \text{ and } w_i \geq 0 \tag{1}$$

where w_i is the component of the representation $\mathbf{w}$ of $\mathbf{p}$. $\mathbf{q}$ is, however, an affine combination of points, as its affine hull is



(a)

(b)

(c)

(d)

Fig. 1. Affine subspace. If $0 \leq w_i \leq 1$, then the point p will be within (or on the boundary) of the triangle. If any w_i is less than zero or greater than 1, then the point will lie outside the triangle. If any w_i is zero, then the point will lie on the boundary of the triangle [43]. (a) Values of p convex hull. (b) Values of q affine hull. (c) Values of p convex hull. (d) Values of q affine hull.

the entire plane [44] [see Fig. 1(b) and (d)]. Formally

$$\mathbf{q} = \sum_{i=1}^{n} w_i \mathbf{x}_i, \quad \text{s.t.,} \quad \sum_{i=1}^{n} w_i = 1. \tag{2}$$

The constraint $\sum_i w_i = 1$ forces the reconstruction of each data point to lie in the subspace spanned by its neighbors; the optimal representations (weights) compute the projection of the data point into this subspace [40]. For simplicity, we consider both Fig. 1(a) and (b). From [43] and [45], it is known that if $0 \leq w_i \leq 1$, then the point p will be within (or on the boundary) of the triangle. If any w_i is less than zero or greater than 1, then the point will lie outside the triangle. If any w_i is zero, then the point will lie on the boundary of the triangle.

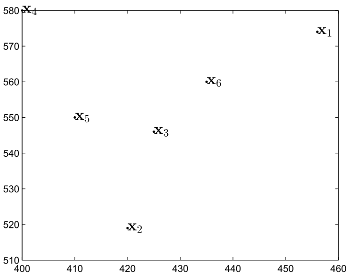


Fig. 2. Simple example. Data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_6\} \in R^{2\times6}$ and the most likely the boundary points are $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4\}$.

Clearly, it has the following.

1) p is within of the triangle if $0 < w_i < 1$.
2) p lies on the boundary of the triangle if any $w_i = 0$.
3) q lies outside the triangle if any $w_i < 0$ or $w_i > 1$.

Note that there are generally simultaneous cases of $w_i > 1$ and $w_i < 0$, $w_i = 1$ and $w_i = 0$ owing to affine combination. The representation $\mathbf{w}$ encodes neighborhood relationships between points in an affine subspace, and it reflects boundary points by negative components [see (8)], as shown at the bottom this page [40]–[42]. For each point, the negative components in its representation generally correspond to the boundary points among its affine combination of points (see Fig. 2) [42]. Thus, data representation $\mathbf{w}$ can uncover the intrinsic data structure [42], [46]. There are many methods that can be used to learn a data representation [30], [40], [47]–[49]. For different purposes, various constraints can be applied to $\mathbf{w}$. Some common constraints are listed in the following:

$$\min \|\mathbf{w}\|_1, \quad \text{s.t.,}\ \mathbf{b} = \mathbf{X}\mathbf{w} \text{ and } \mathbf{1}^T\mathbf{w} = 1 \tag{3}$$

where $\mathbf{w}$ is a column vector [47]

$$\min \sum_{i=1}^{n} |\mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_j|^2, \quad \text{s.t.,} \quad \sum_j W_{ij} = 1 \tag{4}$$

where $\mathbf{W}$ is the representation matrix of $\mathbf{X}$ [40]. Denote

$$\mathbf{X}_i = \left[\frac{\mathbf{x}_1 - \mathbf{x}_i}{\|\mathbf{x}_1 - \mathbf{x}_i\|_2}, \ldots, \frac{\mathbf{x}_n - \mathbf{x}_i}{\|\mathbf{x}_n - \mathbf{x}_i\|_2}\right] \in R^{m\times(n-1)} \tag{5}$$

$$\min\ \lambda_1\|\mathbf{Q}_i\mathbf{w}_i\|_1 + \frac{1}{2}\|\mathbf{X}_i\mathbf{w}_i\|_2^2, \quad \text{s.t.,}\ \mathbf{1}^T\mathbf{w}_i = 1 \tag{6}$$

where λ_1 trades off the sparsity of the solution and the affine reconstruction error, $\mathbf{w}_i$ is a vector, and $\mathbf{Q}_i$ is a proximity inducing matrix. It is a positive-definite diagonal matrix, which favors selecting points that are close to $\mathbf{x}_i$ [30]. Almost all the constraints and their extensions have been widely studied, but only applied in clustering and dimensionality reduction. In the

$$W = \begin{pmatrix} 0 & -0.4024 & 0.1527 & -0.2802 & -0.1176 & 0.4732 \\ -0.5400 & 0 & 0.3718 & -1.0635 & 0.3327 & 0.1147 \\ 0.5400 & 0.9773 & 0 & 0.2748 & 0.2382 & 0.2035 \\ -0.2324 & -0.6584 & 0.0642 & 0 & 0.4315 & 0.1001 \\ -0.3725 & 0.7925 & 0.2156 & 1.6576 & 0 & 0.1084 \\ 1.6048 & 0.2909 & 0.1957 & 0.4112 & 0.1152 & 0 \end{pmatrix} \tag{8}$$

following, assume that inlier data are under a convex set, we discuss how, and under what conditions, the data presentation (w) can be applied to detecting boundary points and outliers.

III. Proposed Method

Since data representation captures sufficient information about the geometry of high-dimensional data, we propose a representation-based method to overcome the limitations of traditional detection methods. In the proposed method, the reverse unreachability of a point is calculated to evaluate to what degree this observation is a boundary point.

A. Data Structure and Negative Components

As shown in [40]–[42], a possible disadvantage of convex approximation (the constraint of nonnegativity) can degrade the reconstruction of data points on the boundary of a manifold and outside the convex of their neighbors. Meanwhile, the negative components (weights) of representation may help to best approximate a point from an affine combination of its neighbors. Therefore, we can make the following remark according to [40]–[42].

Remark 1: The negative components (weights) of points on the boundary of a manifold and outside the convex of their neighbors may be helpful in reducing the reconstruction error in an affine space.

Therefore, negative components can be used to identify boundary points and outliers. However, to assess to what degree each observation is a boundary point or an outlier, an alternative approach is proposed. In general, both types of points are located around the margin of densely distributed data, such as a cluster. More precisely, we state Assumption 1.

Assumption 1: Suppose the set $\mathbf{I}$ includes inner points and $\mathbf{B}$ includes boundary points. In general, the local density of $\mathbf{x}_i \in \mathbf{I}$ is less than that of $\mathbf{x}_j \in \mathbf{B}$.

This assumption is not as restrictive as it might first seem, because the local density of boundary points should be lower than that of inner points generally. From Remark 1 and Assumption 1, we state Remark 2.

Remark 2: The number of negative components of boundary points increases as the distance of a point from its neighbors increases. This indicates that the quantity increases as the distance of a point from the center increases.

It is difficult to conduct thorough quantitative research in theory. To illustrate this, consider the example shown in Fig. 2. The degrees of boundary points, denoted $D(\mathbf{x}_i)$, should increase in the order $\{D(\mathbf{x}_1) \geq D(\mathbf{x}_2) \geq D(\mathbf{x}_4) \geq D(\mathbf{x}_5) \geq D(\mathbf{x}_6) \geq D(\mathbf{x}_3)\}$. Here

$$D(\mathbf{x}_i) = \sum_j \chi(W_{j,i}) \text{ where } \begin{cases} \chi(W_{j,i}) = 1 & \text{if } W_{j,i} < 0 \\ \chi(W_{j,i}) = 0 & \text{otherwise} \end{cases}$$

where $j \in \{1, 2, \ldots, n\}$. From (4), (8) holds,

which uncovers the intrinsic data structure. For $\mathbf{x}_1$, the negative components $\{-0.5400, -0.2324, -0.3725\}$ in its representation generally correspond to the boundary points $\{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5\}$ among its affine combination of points. Clearly, $\{D(\mathbf{x}_1) = 3, D(\mathbf{x}_2) = 2, D(\mathbf{x}_4) = 2, D(\mathbf{x}_5) = 1, D(\mathbf{x}_6) = 0, D(\mathbf{x}_3) = 0\}$, meaning that $\{D(\mathbf{x}_1) > D(\mathbf{x}_2) = D(\mathbf{x}_4) > D(\mathbf{x}_5) > D(\mathbf{x}_6) = D(\mathbf{x}_3)\}$ and showing that Remark 2 is valid.

B. Reverse Reachability

As discussed in Section III-A, negative components can identify boundary points; the number of negative components is used to assess to what degree each observation is a boundary point. However, the data representation is not symmetric, and the set of points with positive components (the nearest neighbors) that are closest to a query point differs from the set of points that have the query point as their nearest neighbor (reverse nearest neighbors) [50], [51]. Therefore, it may be inappropriate to define $D(\mathbf{x}_i)$ only in its local neighborhoods. An alternative method that explicitly takes into account the global data structure and reveals the connectivity between the query point and other points will be proposed. Moreover, the method does not require any precomputation. For simplicity, we describe the method in 2-D space. At the end of the section, we discuss its applicability to arbitrary dimensionality. The method is based on the following definitions.

Definition 1: Given data set $\mathbf{X}$, the "nearest" neighbors of $\mathbf{x}_i$, denoted $\mathcal{NN}(\mathbf{x}_i)$, are defined as

$$\mathcal{NN}(\mathbf{x}_i) = \{\mathbf{x}_j | \forall j \in \{1, 2, \ldots, n\} : W_{j,i} > 0\}.$$

Definition 2: Given data set $\mathbf{X}$, the reverse "nearest" neighbors of $\mathbf{x}_i$, denoted $\mathcal{RNN}(\mathbf{x}_i)$, are defined as

$$\mathcal{RNN}(\mathbf{x}_i) = \{\mathbf{x}_j | \forall j \in \{1, 2, \ldots, n\} : W_{i,j} > 0\}.$$

$\mathcal{RNN}(\mathbf{x}_i)$ may be empty, or have a few elements, especially in the case of boundary points. However, the elements do not directly evaluate to what degree each point is a boundary point. To quantify the number of elements, Definitions 3 and 4 are proposed.

Definition 3: Given data set $\mathbf{X}$, the reachability of $\mathbf{x}_i$, denoted reach$(\mathbf{x}_i)$, is the number of points that $\mathbf{x}_j \in \mathbf{X}$ with $W_{j,i} > 0$, $j \in \{1, 2, \ldots, n\}$. Formally

$$\text{reach}(\mathbf{x}_i) = \sum_j \chi(W_{j,i}) \text{ where } \begin{cases} \chi(W_{j,i}) = 1 & \text{if; } W_{j,i} > 0 \\ \chi(W_{j,i}) = 0 & \text{otherwise.} \end{cases}$$

Definition 4: Given data set $\mathbf{X}$, the reverse reachability of $\mathbf{x}_i$, denoted Rreach$(\mathbf{x}_i)$, is the number of points that $\mathbf{x}_j \in \mathbf{X}$ with $W_{i,j} > 0$, $j \in \{1, 2, \ldots, n\}$. Formally

$$R\text{reach}(\mathbf{x}_i) = \sum_j \chi(W_{i,j}) \text{ where } \begin{cases} \chi(W_{i,j}) = 1 & \text{if } W_{i,j} > 0 \\ \chi(W_{i,j}) = 0 & \text{otherwise.} \end{cases}$$

Rreach$(\mathbf{x}_i)$ has properties that are uniquely different from those of the conventional reach$(\mathbf{x}_i)$.

1) Rreach$(\mathbf{x}_i)$ is not localized to the neighborhood of $\mathbf{x}_i$.
2) Rreach$(\mathbf{x}_i)$ reflects the data structure, i.e., Rreach$(\mathbf{x}_i)$ varies according to the data distribution [52].

From $\mathbf{W}$ for Fig. 2, we see that Rreach$(\mathbf{x}_1) = 2$, Rreach$(\mathbf{x}_2) = 3$, Rreach$(\mathbf{x}_3) = 5$, Rreach$(\mathbf{x}_4) = 3$, Rreach$(\mathbf{x}_5) = 4$, and Rreach$(\mathbf{x}_6) = 5$. Note that $\mathbf{x}_3$ and

$\mathbf{x}_6$ each have five reverse "nearest" neighbors, while $\mathbf{x}_1$ has two reverse "nearest" neighbors. The larger the distance of a point from other points (not only its local neighbors), the smaller the value of Rreach. These properties of Rreach have potential application in data mining, such as in the detection of boundary points and outliers.

C. Reverse Unreachability

Conversely, on the basis of the above discussion, the reverse unreachability (denoted by $\mathcal{NC}$) is adopted to identify boundary points. For each point $\mathbf{x}_i, i \in \{1, 2 \ldots, n\}$, all the points $\mathbf{x}_j \in \mathbf{X}$ that have $\mathbf{x}_i$ as a boundary or irrelevant neighbor and do not require any precomputation are retrieved. Formally

$$\mathcal{NC}_i = \sum_j \chi(W_{i,j}), \quad j \in \{1, 2, \ldots, n\} \tag{7}$$

where $\chi(W_{i,j}) = 1$ if $W_{i,j} \leq 0$ and $\chi(W_{i,j}) = 0$ otherwise. Note that $\mathcal{NC}_i$, which simultaneously considers points that have $\mathbf{x}_i$ as the irrelevant point ($W_{i,j} = 0$) and boundary neighbor ($W_{i,j} < 0$), is not localized to the neighborhood of $\mathbf{x}_i$. Rreach($\mathbf{x}_i$) reflects the reverse "nearest" neighbors for each point $\mathbf{x}_i$, while $\mathcal{NC}_i$ first explicitly takes into account the structure of the data on which they may possibly reside, and simultaneously considers points that have $\mathbf{x}_i$ as irrelevant and boundary points. More importantly, $\mathcal{NC}_i$ provides a new metric that can be used to assess to what degree each point is a boundary point. This is more interesting and obviously preferable. Compared with other related methods, our method better reflects the characteristics of data.

The same observations show that (7) can be applied to outlier detection. More generally, the $\mathcal{NC}_i$ value of outliers is larger than that of boundary points, allowing the retrieval of only the top-m ranked points. The following definition of an outlier is then proposed.

Definition 5: Given data $\mathbf{X}$, an outlier is a point with a high $\mathcal{NC}_i$ value. The higher the value of $\mathcal{NC}_i$, the more likely it is that the point is an outlier.

Note that the score of reverse unreachability of an outlier is greater than that of a boundary point. The top-m ranked points can thus be identified as outliers. Except for the top-m outliers, it states that the greater the value of the reverse unreachability, the more likely the point is a boundary point.

D. Computational Complexity

Let $\mathcal{NC}_i$ denote the ith element of the vector $\mathcal{NC}$. By (7), the time complexity of our algorithm is based on the following procedure. Clearly, the time complexity is $O(n^2)$. $\mathbf{W}$ is generally constructed with a closed-form solution.

```
Compute $\mathcal{NC}_i$ ($\mathbf{W}$)
{ $\mathcal{NC} = zeros(n, 1)$
for $i = 1$ to $n$ do
    for $j = 1$ to $n$ do
        $\mathcal{NC}_i = \mathcal{NC}_i + \chi(W_{i,j})$
    end for
end for}
```

Algorithm 1 Boundary and Outlier Detection Algorithm

```
Input: $\mathbf{X} \in R^{d \times n}$, $k$
Output: top-$m$ points
Step 1: Construct weight matrix $\mathbf{W}$ via Eq.(4)
Step 2: Compute $\mathcal{NC}_i$ via (7)
Step 3: Rank $\mathcal{NC}_i$
```

E. Extensions

1) Identify k: To learn a data representation, an appropriate function can be chosen for a given application. A proper choice of the neighborhood size k is important. To handle this issue, (6) can be employed in a technique that allows the robust automatic selection of neighbors [30]. Initialize k with a large value, both the neighbors and weights are found automatically [30]. In general, we can set $k = n - 1$ for a single class or manifold, and an appropriate value increasing with the size of data set for multiple classes or manifolds (such as $k = 6\log(n)$ like [22]). Note that the global structure and not local is considered, the value of k is not critical in our method.

2) Data Reduction: Data reduction is the process of transforming masses of data into a small number of summarized reports. The proposed method can reduce the size of training data and eliminate instances. Keeping only a subset of training data for online classification reduces storage requirements and improves the search time and memory requirements.

The overall method is summarized in Algorithm 1. If (6) is employed to build $\mathbf{W}$, k is initialized with a large value [30]. The ranked $\mathcal{NC}_i$ values can be used in many applications, such as the detection of boundary points and outliers and data reduction. This paper focuses on the detection of boundary points and outliers.

IV. Experiments and Discussion

A. Experimental Setup

For outlier detection, we evaluate the performance of the related methods, including k-nearest neighbor (kNN), ABOD, and FastABOD [21]. In all the experiments, kNN defines the distance of a given data point to its kth nearest neighbors as the outlier or boundary score [11]. The greater the value of the score, the most likely the boundary point. Its time complexity is $O(n^2)$ and identifies the top-m ranked points as outlier points. The time complexity of ABOD is in $O(n^3)$, while that of FastABOD is in $O(n^2 + n \cdot k^2)$, where k is the number of nearest neighbors [21]. Basic metrics, precision (P) and recall (R), were used to evaluate the ability of each algorithm to retrieve the most likely outlier. Formally, we denote a = Card{*relevant records retrieved*}, b = Card{*relevant records not retrieved*}, and c = Card{*irrelevant records retrieved*}, and then $P = (a/(a + c)) \cdot 100\%$ and $R = (a/(a + b)) \cdot 100\%$. Let m = the number of outliers, then $c = m - a$ and $b = m - a$. Thus, $P = R$, denoted by P(R), in the top-m retrieving step. To plot precision and recall graph (Receiver Operating Characteristic curve), let $m = 1, 2, \ldots,$ the number of outliers.

In all the experiments, we highlight the positions of the $n \cdot 10\% - n \cdot 20\%$ points with greater scores for kNN. The best results are reported with tuned parameter k for kNN. BEPS using cross-validation approach can select two kinds of critical patterns: border points and edge points. Border points are patterns between classes, while an edge point is a pattern sites outmost around a data set (e.g., class, whole data). Moreover, an edge point must not be a border points [22]. There are four parameters in BEPS. In general, set $k_b = \text{round}(5{\cdot}\log10(n)/2){\cdot}2{+}1$, $k_e = \text{round}(5{\cdot}\log(n)/2){\cdot}2{+}1$, $\lambda = 70$ and $\gamma = 90$ unless otherwise stated [22]. Parameters k_b and λ are used for computing border points (e.g., black circles in Fig. 5) and k_e and γ for computing edge points (e.g., red diamonds in Fig. 5). Please refer to this source for the details.

For the sake of simplicity of computation, (4) with a closed-form solution is generally used to construct $\mathbf{W}$ in our method. We typically set $k = \text{round}(6\log(n))$. Note that generally the positions of about $n{\cdot}10\%{-}n{\cdot}20\%$ points with bigger $\mathcal{NC}_i$ than certain value are highlighted for boundary detection and the top-m ranked points as outliers. Their performance is assessed on several data sets with different characteristics, such as data dimension, class balance, and noise content. The program for BEPS was provided by its originator. All experiments were written in MATLAB and run on an Intel Pentium Dual 2.2 GHz CPU E2200 with 4 GB (3.25 GB available) main memory.

B. Validation

To evaluate the performance of boundary detection, we compare our method with kNN and BEPS methods. BEPS is evaluated on benchmark problems using popular classifiers [22]. The main code for BEPS was provided by their respective originators. Both attempt to select critical points that preserve the decision surface of the original data set. Furthermore, the effectiveness of boundary detectors must exert impact on the performance of clusters, but not vice versa. Thus, we first study the effectiveness of boundary detectors by showing the relative locations of boundary points identified on the different data sets with various distributions and different numbers of clusters. Moreover, we can further validate the effectiveness of boundary detectors with cluster performance by combining clusters. We selected the two most popular clustering algorithms: k-means and SMCE [53], which do not involve training.

C. Visualization Results

While boundary detection methods can be used to other applications, the criteria may be unsuitable for evaluating the performance. Moreover, there are no standards to determine boundary quality in a data set. The simple way to illustrate the performance of boundary detection methods is to show the relative locations of boundary points in visible space.

1) XOR Data: Like [22], XOR data ($\mathbf{X} \in R^{2\times2000}$) were generated using normal distribution $N(\mu, \sigma^2)$, where μ is the mean vector and σ^2 is the variance. Set $\sigma = 0.5$, and mean vectors (1,1) and (−1,−1) for class one and (−1,1) and (1,−1) for class two (each class including two clusters). Note that BEPS needs one thousand points for the training data and 1000 for test. Like BEPS, set $k = \text{round}(5 \cdot \log(n)/2) \cdot 2 + 1$ for kNN.

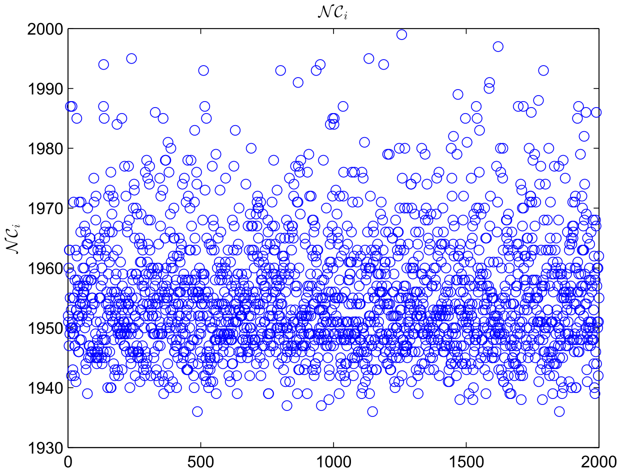


Fig. 3. $\mathcal{NC}_i$ of the XOR data by our method.

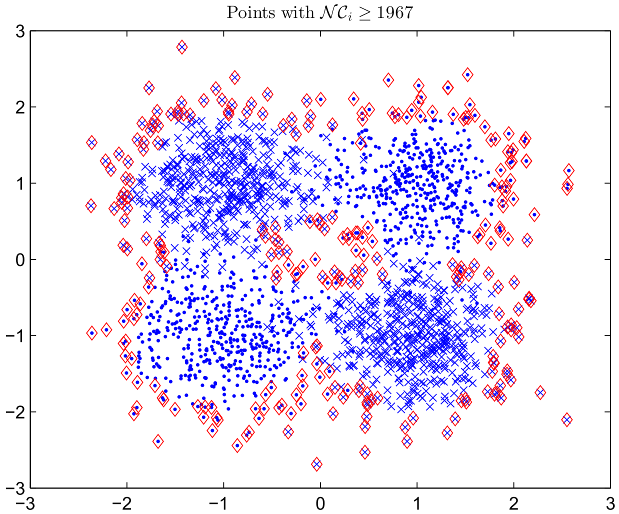


Fig. 4. XOR data set ($\mathbf{X} \in R^{2\times2000}$), dot for class 1, cross for class 2. Red diamonds represent boundary points by our method. These red diamonds are corresponding points with $\mathcal{NC}_i \geq 1967$.

The corresponding $\mathcal{NC}_i$ values by our method are shown in Fig. 3. Both horizontal and vertical axes correspond to the ith point and its corresponding $\mathcal{NC}_i$ value, respectively. About $2000{\cdot}10\%$ points with $\mathcal{NC}_i \geq 1967$ are marked (red diamonds) in Fig. 4. Fig. 5 shows the critical points (border points and edge points) by BEPS. Fig. 6 shows the top $2000 \cdot 10\%$ points as boundary points by kNN. One can see that these selected points (red diamonds) by our method are located near the margin of the XOR data. BEPS gets a better performance than kNN, while it is obvious that some critical points (i.e., points in rectangle in Fig. 5) are not identified. Moreover, BEPS has four parameters. Compared with BEPS and kNN, our method better reflects the characteristics of the data.

2) Simple Examples for Boundary and Outlier Detection: The core of the algorithm is illustrated by simple examples

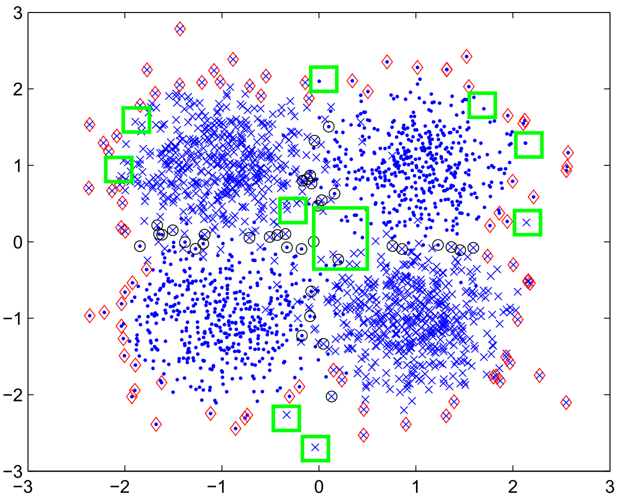


Fig. 5. XOR data set, dot for class 1, cross for class 2. Red diamonds represent edge points and black circles represent border points by BEPS.

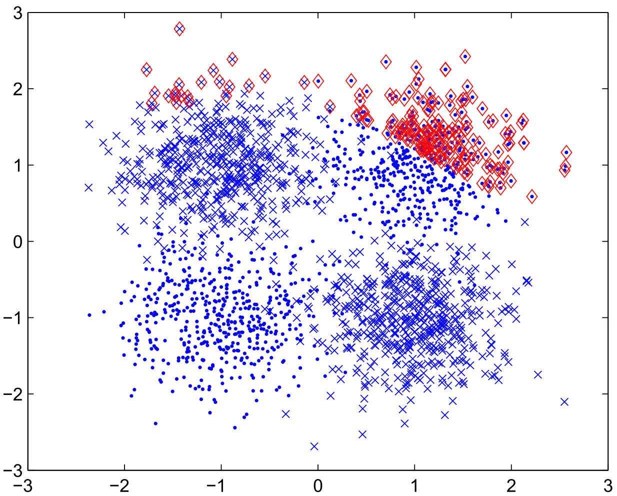


Fig. 6. XOR data set ($\mathbf{X} \in R^{2\times 2000}$), dot for class 1, cross for class 2. Red diamonds represent boundary points by kNN.

(see Figs. 7 and 9). Fig. 7(b) shows the Hertzsprung–Russell data of the star cluster CYG OB1 in 2-D space. The first attribute is the logarithm of the effective temperature of the surface of a star and the second is the logarithm of the light intensity of the star [13]. Fig. 7(a) shows the values of $\mathcal{NC}_i$ for each point; we refer to this representation as the score graph. Both horizontal and vertical axes correspond to the ith point and its corresponding $\mathcal{NC}_i$ value, respectively. According to Definition 5, $\{\mathcal{NC}_{34}, \mathcal{NC}_{30}, \mathcal{NC}_{20}, \mathcal{NC}_{11}, \mathcal{NC}_{17}\}$ are slightly smaller than $\{\mathcal{NC}_7, \mathcal{NC}_{14}\}$. Fig. 7(a) shows that $\mathbf{x}_7$ and $\mathbf{x}_{14}$ with higher scores are the most likely outliers. Table II shows the top-two-ranked points and corresponding precision. Fig. 8 shows the precision and recall graph with $m = 1, 2$. Our method performs better than other related methods (see Table II). To demonstrate the detection of boundary points, the top-8 and top-14 ranked points are shown in Fig. 7(c) and (d), respectively. It is seen that the points with higher scores are located near the margin of data. Furthermore, some points with lower $\mathcal{NC}_i$ values can be eliminated to reduce storage requirements and improve the search time and memory requirements.

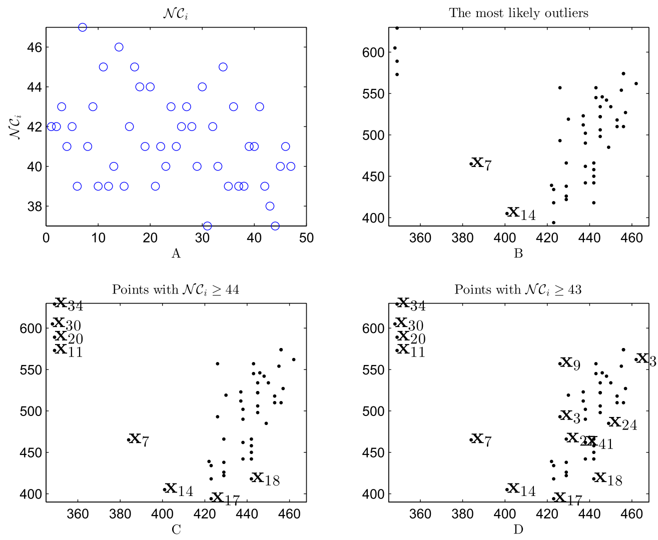


Fig. 7. (a) $\mathcal{NC}_i$ of the data in (b). (b) Hertzsprung–Russell data of the star cluster CYG OB1. (c) and (d) Points with $\mathcal{NC}_i \geq 44$ and $\mathcal{NC}_i \geq 43$.

TABLE II

TOP-TWO-RANKED POINTS AND CORRESPONDING PRECISION ACHIEVED USING EACH METHOD FOR THE HERTZSPRUNG–RUSSELL DATA SET (SET $k = 4$ FOR FastABOD AND $k = 10$ FOR kNN)

Method	*top-2-ranked points*	P(%)
ABOD	$\mathbf{x}_{34}, \mathbf{x}_7$	50.00
FastABOD	$\mathbf{x}_7, \mathbf{x}_{34}$	50.00
kNN	$\mathbf{x}_{19}, \mathbf{x}_{14}$	50.00
$\mathcal{NC}_i$	$\mathbf{x}_7, \mathbf{x}_{14}$	100.00

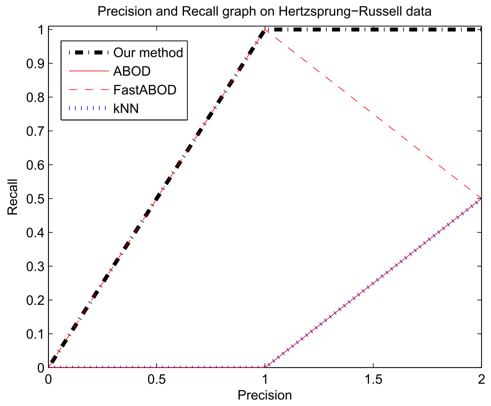


Fig. 8. Precision and recall graph on Hertzsprung–Russell data. Let $m = 1, 2$.

To benchmark our procedure, let us first consider the test case in Fig. 9. Fig. 9(b) shows FLAME data [54], [55] in a 2-D space. It is seen that $\mathbf{x}_1$ and $\mathbf{x}_2$ are the two most likely outliers. The corresponding $\mathcal{NC}_i$ scores are shown in Fig. 9(a). We see that $\mathbf{x}_1$ and $\mathbf{x}_2$ have large $\mathcal{NC}_1$ and $\mathcal{NC}_2$ values, respectively. According to Definition 5, the two points are identified as the most likely outliers. Although ABOD and FastABOD identify both outliers, they do not work well on detecting boundary points. kNN does not identify both outliers.

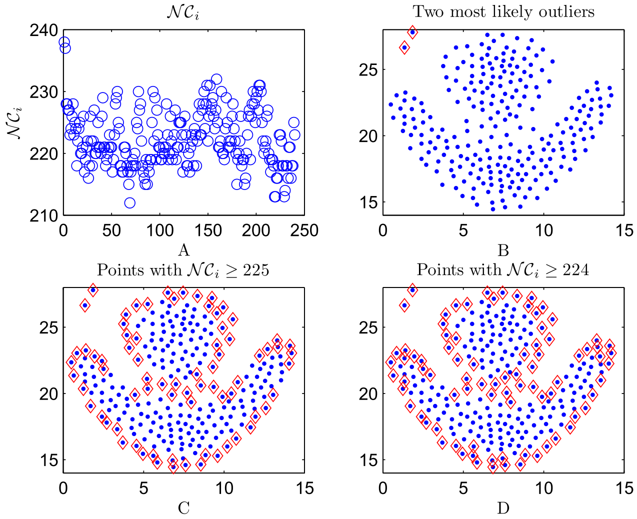


Fig. 9. (a) $\mathcal{NC}_i$ of the FLAME in (b). (b) FLAME data ($\mathbf{X} \in R^{2\times788}$). (c) and (d) Points with $\mathcal{NC}_i \geq 225$ and $\mathcal{NC}_i \geq 224$.

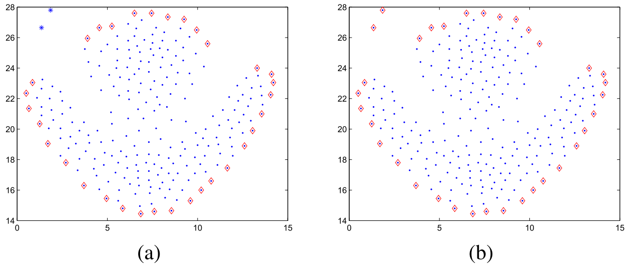


Fig. 10. FLAME data ($\mathbf{X} \in R^{2\times788}$). (a) Dot for class one, star for class 2. (b) Take FLAME data as a class. Red diamonds represent edge points and black circles represent border points by BEPS.

To demonstrate the effectiveness of detection of boundary points, Fig. 9(c) and (d) shows the top points with $\mathcal{NC}_i \geq 225$ and $\mathcal{NC}_i \geq 224$, respectively. It is clear that these points are located near the margin of the FLAME data. Since BEPS using cross-validation approach two experiments have been done. When BEPS takes FLAME data as two classes, the selected critical points are shown in Fig. 10(a). Taking FLAME data as one class, BEPS can identify more edger points [see Fig. 10(b)]. Nevertheless, in both cases, some border and edge points are not identified. Fig. 11(a) shows boundary points by kNN. About $n \cdot 20\%$ points are selected as boundary points. Compared with the related methods, our method better reflects the characteristics of the data, and simultaneously detects outliers and boundary points. Conversely, some points with lower $\mathcal{NC}_i$ can be eliminated to reduce storage requirements and improve the search time and memory requirements.

We next verify the effectiveness of the proposed method for the case shown in Fig. 12. The same conclusion drawn for the previous two data sets can be reached for the Aggregation data set [56] [see Figs. 11(b), 12, and 13]. Fig. 12 shows points with $\mathcal{NC}_i \geq 765$. One can see that the annotated points are located near the margin of the Aggregation data by our method. Compared with the related methods [see Figs. 11(b) and 13], our method better reflects the characteristics of Aggregation data.

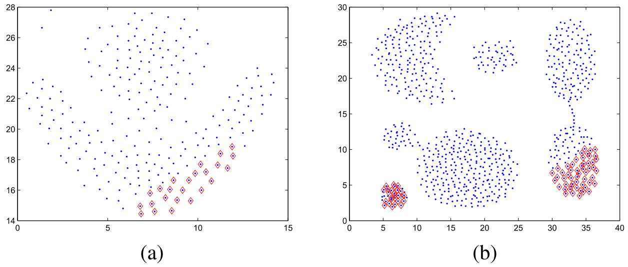


Fig. 11. Points with $\mathbf{x}_i$ are selected boundary points by kNN. Setting $k = \text{round}(5 * \log(n)/2) * 2 + 1$. (a) FLAME data. (b) Aggregation data.

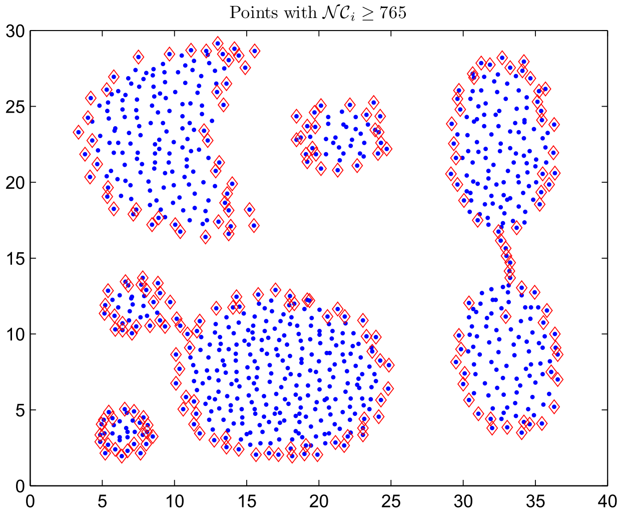


Fig. 12. Points with $\mathcal{NC}_i \geq 765$ in Aggregation data set.

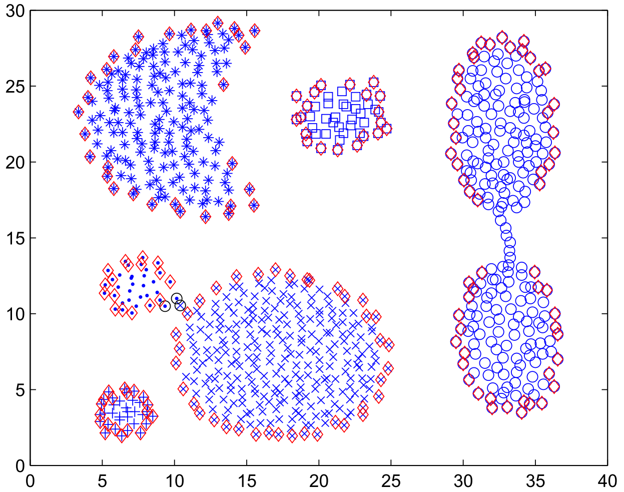


Fig. 13. Aggregation data with different class symbols. Red diamonds represent edge points and black circles represent border points by BEPS. Note that only three border points are identified between several classes.

3) BUPA Database: The BUPA Liver Disorders data set ($X \in R^{7\times345}$) was obtained from the UCI Machine Learning Database Repository [57]. The first five attributes denote the results of different blood tests, which are thought to be sensitive to liver disorders, while the seventh attribute is a selector field used to split the data into two sets with 145 and 200 instances. Note that we abbreviate $\mathbf{x}_i$ as i for the sake of

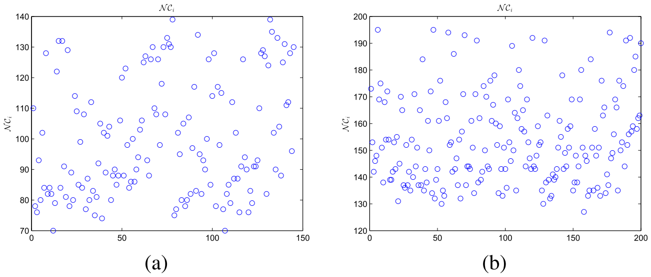


Fig. 14. (a) $\mathcal{NC}_i$ for the first class of the BUPA data. Setting $k = 70$. (b) $\mathcal{NC}_i$ for the second subset of BUPA data. Setting $k = 70$.

TABLE III

NORMAL POINTS RETRIEVED AND CORRESPONDING RECALL IN THE TOP-m STEP ON THE BUPA DATA SET. SETTING $k = 8$ FOR FastABOD AND $k = 4$ FOR kNN

Method	class	*Normal points*	R(%)
ABOD	1 2	*172, 329* *334, 151, 98, 53*	90.91 84.62
FastABOD	1 2	*19, 172, 203* *334, 151, 41, 53, 220, 83*	86.36 76.92
kNN	1 2	*27, 30, 106, 172, 315* *53, 98, 102, 128, 151, 155, 277, 334*	77.27 69.23
$\mathcal{NC}_\mathrm{i}$	1 2	*13, 214* *normal points:* 38, 43, 83, 98, 154, 321, 334 *among points with* $\mathcal{NC}_\mathrm{i} \geq 172$	90.91 96.15
	2	*normal points:* 38, 43, 83, 98, 154, 321, 334, 63, 122 *among points with* $\mathcal{NC}_\mathrm{i} \geq 171$	100.00

clarity of notation. As shown in [58], the doubtful outliers in the two classes are as follows.

1) *Outliers in Class* 1 *(22): 168, 175, 182, 190, 205, 316, 317, 335, 345, 148, 183, 261, 311, 25, 167, 189, 312, 326, 343, 313, 20, 22.*
2) *Outliers in Class* 2 *(26): 36, 77, 85, 115, 134, 179, 233, 300, 323, 331, 342, 111, 139, 252, 294, 307, 123, 186, 286, 2, 133, 157, 187, 224, 278, 337.*

The corresponding $\mathcal{NC}_i$ scores of the first class are shown in Fig. 14(a). We list points with $\mathcal{NC}_i \geq 127$.

1) *Points With* $\mathcal{NC}_i \geq 127$ *in Class* 1 *(22): 13, 20, 22, 25, 148, 168, 175, 182, 183, 189, 190, 205, 214, 311, 312, 313, 316, 317, 326, 335, 343, 345.*

Comparison with outliers in class 1 shows that the precision and recall is 90.91% in the top-22 retrieval step (see Table III). According to the definition of outliers and boundary points [14], [22], both 13 and 214 (not outliers) among points with $\mathcal{NC}_i \geq 127$ may be boundary points, while it is impossible to annotate the margin or outlier points in higher dimensional visualizations. To illustrate this, we select the subset of attributes, {1, 3, 5} employing the sequential forward selection method [58] and annotate points with $\mathcal{NC}_i \geq 127$ in the 3-D space [see Fig. 15(a)]. Fig. 15(a) shows that the three attributes mainly reflect the BUPA data distribution; both points 13 and 214 are boundary points and the outliers are located near the margin of the data set. Compared with other related methods, our method performs better in outlier detection (see Table III and Fig. 16).

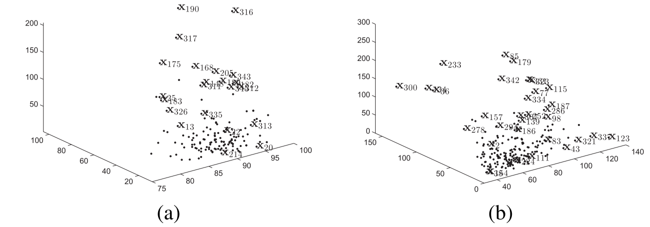


Fig. 15. (a) Locations of points with $\mathcal{NC}_i \geq 127$ in 3-D space. (b) Locations of points with $\mathcal{NC}_i \geq 172$ in 3-D space.

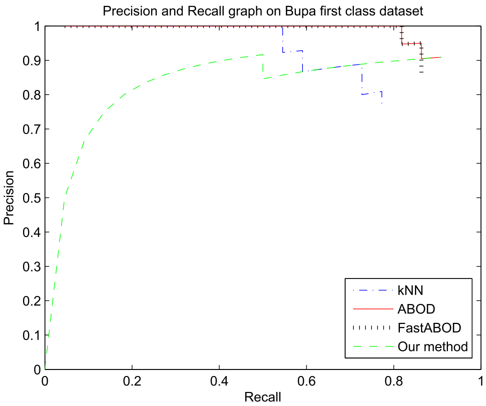


Fig. 16. Precision and recall graph on Bupa first data set. Let $m = 1, \ldots, 22$.

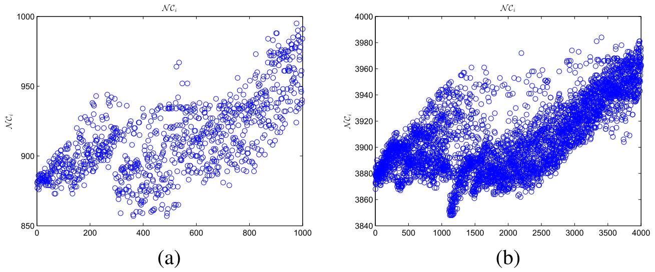


Fig. 17. Corresponding $\mathcal{NC}_i$. Setting $k = 100$. (a) Value of $\mathcal{NC}_i$ of PanelC. (b) Value of $\mathcal{NC}_i$ of PanelB.

Fig. 14(b) shows the corresponding $\mathcal{NC}_i$ scores of the second class. We list points with $\mathcal{NC}_i \geq 172$ and $\mathcal{NC}_i \geq 171$.

1) *Points With* $\mathcal{NC}_i \geq 172$ *in Class 2: 2, 36, 38, 43, 77, 83, 85, 98, 111, 115, 123, 134, 139, 154, 157, 179, 186, 187, 224, 233, 252, 278, 286, 294, 300, 307, 321, 323, 331, 334, 337, 342.*
2) *Points With* $\mathcal{NC}_i \geq 171$ *in Class 2:* {*63, 122, 133*} $\bigcup$ {points with $\mathcal{NC}_i \geq 172$}.

The comparison with outliers in class 2 shows that the recall is 96.15% among points with $\mathcal{NC}_i \geq 172$. Note that only {133} cannot be found among points with $\mathcal{NC}_i \geq 172$. Fig. 16 shows the corresponding precision and recall graph. Furthermore, it can achieve 100% recall for points with $\mathcal{NC}_i \geq 171$. To mainly reflect the distribution of outliers in BUPA data, the subset of attributes {2, 3, 5} is selected. Fig. 15(b) shows that the three attributes mainly reflect the distribution of the second class; the outliers and other points {38, 43, 83, 98, 154, 321, 334} are located near the margin of the data.

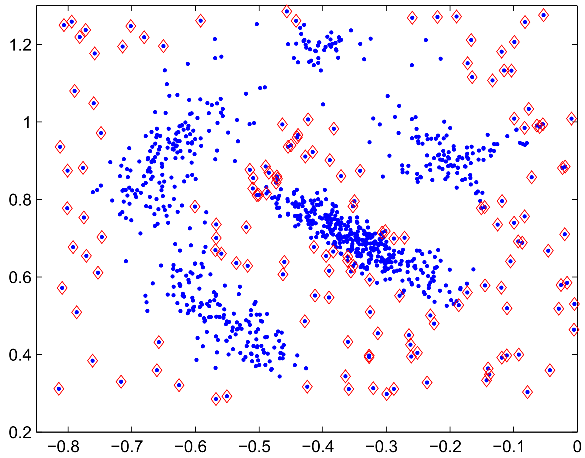


Fig. 18. Annotated PanelC points with $\mathcal{NC}_i \geq 938$. Setting $k = 100$.

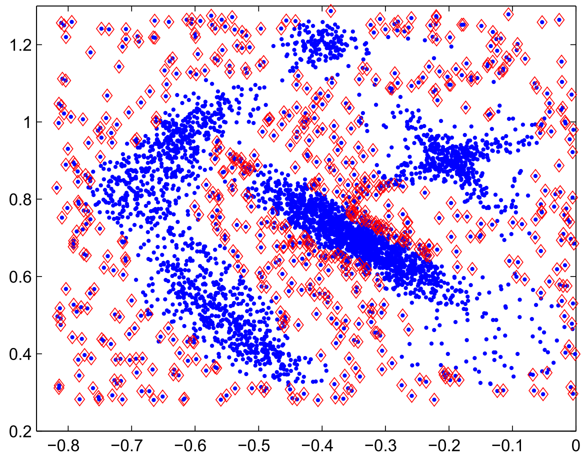
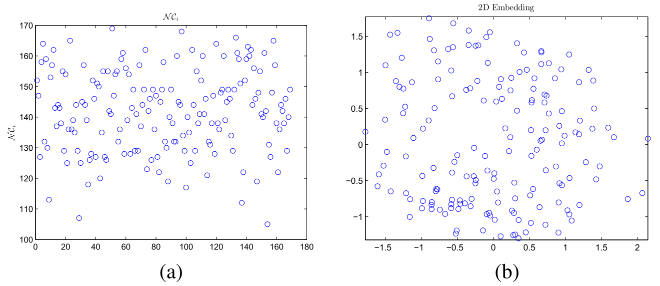


Fig. 19. Annotated PanelB points with $\mathcal{NC}_i \geq 3940$. Setting $k = 100$.

4) Noise Data: In Figs. 18 and 19, 1000 and 4000 points, respectively, are drawn from the same probability distribution, and cluster centers are surrounded by neighbors with lower local density [55]. The corresponding $\mathcal{NC}_i$ scores are shown in Fig. 17(a) and (b), respectively. Both horizontal and vertical axes correspond to the ith point and its corresponding $\mathcal{NC}_i$ value, respectively. The two figures show that the $\mathcal{NC}_i$ values are similar for the two selections and differ only in terms of cardinality. Fig. 18 shows PanelC points with $\mathcal{NC}_i \geq 938$, and Fig. 19 shows PanelB points with $\mathcal{NC}_i \geq 3940$. Both figures illustrate that the annotated points with higher $\mathcal{NC}_i$ values are located near the margin of the clusters. This is useful in removing noise that is unpredictable in data.

5) MNIST Database: To demonstrate clearly the semantic meaning behind the method, a real MNIST database is employed. We use 169 sampled images of five handwritten digits to evaluate the proposed method in the detection of boundary points. Note that each image is of size 28×28 pixels, depicting a point in a 784-D space.



(a) (b)

Fig. 20. Results for MNIST data set. (a) Value of $\mathcal{NC}_i$. (b) 2-D embedding of MNIST.

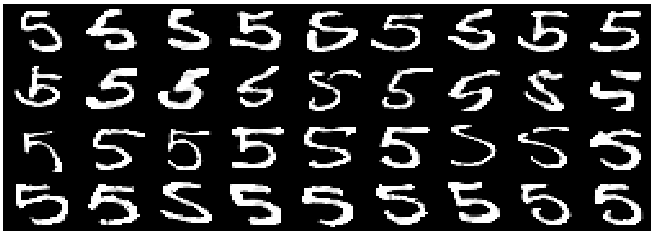


Fig. 21. Images with $\mathcal{NC}_i \geq 155$. Note that there are 37 images with $\mathcal{NC}_i \geq 155$, we select 36 points to facilitate image display (from left to right and from top to bottom).

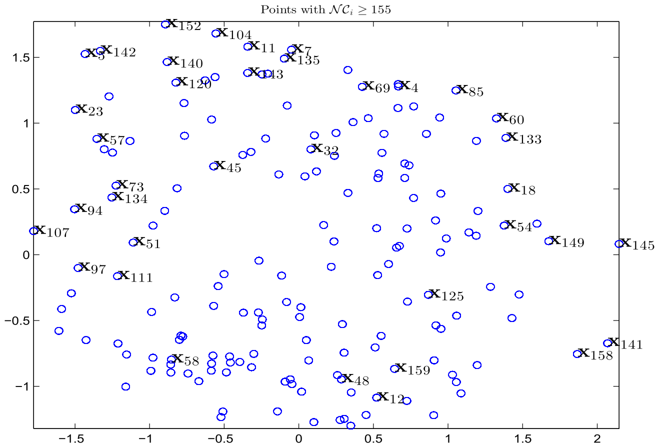


Fig. 22. Annotated points with $\mathcal{NC}_i \geq 155$ in 2-D embedding space.

Owing to the complexity of the real data, (6) is employed to construct $\mathbf{W}$. We set $\lambda_1 = 0.1$ and initialize $k = 50$. Fig. 20(a) shows the value of $\mathcal{NC}_i$. Fig. 21 shows the images corresponding to points with $\mathcal{NC}_i \geq 155$. It is seen that these images are different from each other in that there is a vertical change between the images. However, we do not directly know whether the points are near the margin of the data set in high-dimensional spaces. To solve the problem, the 2-D embedding of the data is shown in Fig. 20(b). Fig. 22 shows the points with $\mathcal{NC}_i \geq 155$ corresponding to images in Fig. 21. Although the data distribution does not meet Assumption 1 strictly, it is clear that these points are generally located near the margin of the data set.

D. Cluster Performance

The effectiveness of boundary detectors must exert impact on the performance of a cluster, but not vice versa. However, better boundary detectors with better clusters would get better results. Since the performance of the latter in general depends greatly on the strategy employed by the cluster, we carry

TABLE IV

CLUSTERING PRECISION ACHIEVED USING EACH METHOD FOR THE FIRST TYPE OF BANANA DATA SET

Method	♯ REMOVED BOUNDARY POINTS	CLUSTER	P(%)
		KMEANS	97.30
BEPS	193	KMEANS	98.24
$\mathcal{NC}_{\rm i}$	201	KMEANS	98.80
		SMCE	99.97
BEPS	193	SMCE	100.00
$\mathcal{NC}_{\rm i}$	218	SMCE	100.00

TABLE V

CLUSTERING PRECISION ACHIEVED USING EACH METHOD FOR THE WINE DATA SET

Method	♯ REMOVED BOUNDARY POINTS	CLUSTER	P(%)
		KMEANS	70.22
BEPS	43	KMEANS	68.89
$\mathcal{NC}_{\rm i}$	31	KMEANS	72.11
		SMCE	70.22
BEPS	43	SMCE	62.96
$\mathcal{NC}_{\rm i}$	31	SMCE	73.47

out some experiments on boundary detectors combined with the same cluster detectors to illustrate the role of boundary detectors.

1) Banana Data: We obtained the Banana data set ($X \in R^{3\times 5300}$) from the IDA Benchmark repository. We can split the data into two sets with 2924 and 2376 instances by the third selector field. To illustrate the role of boundary detectors, we divide the first type of banana data set into two categories. If the k-means algorithm directly works on the first type of banana data, then there exist some misclassified points.

It can be seen that some boundary points identified by our method are hard to classify, and all the rest are clustered correctly by k-means. Table IV shows the precision achieved using each pattern selection method for the first type of banana data set. We can see that the better boundary detection method combined with the same clustering algorithm would perform better. It also illustrates that the effectiveness of boundary detectors must exert impact on the performance of the cluster, but not vice versa.

2) Wine Recognition Data: The wine recognition data set ($X \in R^{13\times 178}$) contains multivariate data. It was updated September 21, 1998, by C. Blake: added attribute information. It consists of 178 samples from each of three kinds of wines with 13 features: 1) alcohol; 2) malic acid; 3) ash; 4) alcalinity of ash; 5) magnesium; 6) total phenols; 7) flavonoids; 8) non-flavonoid phenols; 9) proanthocyanidins; 10) color intensity; 11) hue; 12) OD280/OD315 of diluted wines; and 13) proline. There are 59 instances of class 1, 71 of class 2, and 48 of class 3. Table V illustrates the precision achieved using each pattern selection method for the wine data set. Compared with the BEPS method, our method has a better performance when combined with k-means or SMCE clustering algorithms. It can be seen that each method can achieve high precision after removing boundary points.

V. CONCLUSION

This paper presented an efficient representation-based method that can be used to simultaneously identify both outliers and boundary points, regardless of their distribution and the dimensionality of the space. In the presented method, the reverse unreachability of a point is proposed to evaluate to what degree this observation is a boundary point. It can be calculated by counting the number of zero and negative components in the representation. The reverse unreachability explicitly takes into account the global data structure and reveals the disconnectivity between a data point and other points. This paper reveals that the reverse unreachability of points with lower density has a higher score than that of points with high density. Note that the score of reverse unreachability of an outlier is greater than that of boundary points. The top-m ranked points are identified as outliers, which are more interesting and obviously preferable for further investigation. The greater the value of the reverse unreachability, the most likely the point is a boundary point. Compared with related methods, our method better reflects the characteristics of the data and simultaneously detects outliers and boundary points regardless of their distribution and the dimensionality of the space. Experimental results on a number of synthetic and real-world data sets demonstrate the effectiveness and efficiency of our method.

REFERENCES

[1] J. C. Lv, K. K. Tan, Z. Yi, and S. Huang, “A family of fuzzy learning algorithms for robust principal component analysis neural networks,” *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, pp. 217–226, Feb. 2010.

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *Adv. Knowl. Discovery Data Mining*, vol. 17, no. 3, pp. 37–54, 1996.

[3] V. S. Tseng, T. B. Ho, Z. Zhou, A. L. P. Chen, and H. Kao, Eds., *Advances in Knowledge Discovery and Data Mining* (Lecture Notes in Artificial Intelligence), vol. LNAI 8444. Springer International: Springer, 2014.

[4] J. C. Lv, Z. Yi, and J. Zhou, *Subspace Learning of Neural Networks*. Boca Raton, FL, USA: CRC Press, 2011.

[5] J. C. Lv, Z. Yi, and K. K. Tan, “Determination of the number of principal directions in a biologically plausible PCA model,” *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 910–916, May 2007.

[6] X. Li, J. C. Lv, and D. Cheng, “Angle-based outlier detection algorithm with more stable relationships,” in *Proc. 18th Asia Pacific Symp. Intell. Evol. Syst.*, vol. 1. 2014, pp. 433–446.

[7] D. M. Hawkins, *Identification of Outliers*, vol. 11. New York, NY, USA: Springer, 1980.

[8] C. C. Aggarwal and P. S. Yu, “Outlier detection for high dimensional data,” in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2001, pp. 37–46.

[9] L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady, “Novelty detection for the identification of masses in mammograms,” in *Proc. 4th Int. Conf. Artif. Neural Netw.*, Jun. 1995, PP. 442–447.

[10] H.-P. Kriegel, P. Kröger, and A. Zimek, “Outlier detection techniques,” in *Proc. 13th Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2009, pp. 1–73.

[11] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.

[12] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artif. Intell. Rev.* vol. 22, no. 2, pp. 85–126, 2004.

[13] V. Hautamäki, I. Kärkkäinen, and P. Fränti, “Outlier detection using k-nearest neighbour graph,” in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3. 2004, pp. 430–433.

[14] C. C. Aggarwal and P. S. Yu, “Outlier detection for high dimensional data,” *ACM SIGMOD Rec.*, vol. 30, no. 2, pp. 37–46, 2001.

[15] T. Kutsuna and A. Yamamoto, “Outlier detection based on leave-one-out density using binary decision diagrams,” in *Advances in Knowledge Discovery and Data Mining*. Springer International: Springer, 2014, pp. 486–497.

[16] G. Ratsch, S. Mika, B. Scholkopf, and K. R. Muller, "Constructing boosting algorithms from SVMs: An application to one-class classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1184–1199, Sep. 2002.

[17] Y.-H. Liu, Y.-C. Liu, and Y.-J. Chen, "Fast support vector data descriptions for novelty detection," *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1296–1313, Aug. 2010.

[18] X. Peng and D. Xu, "Efficient support vector data descriptions for novelty detection," *Neural Comput. Appl.*, vol. 21, no. 8, pp. 2023–2032, 2012.

[19] S. Byers and A. E. Raftery, "Nearest-neighbor clutter removal for estimating features in spatial point processes," *J. Amer. Statist. Assoc.*, vol. 93, no. 442, pp. 577–584, 1998.

[20] M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. Int. Conf. ACM SIGMOD Manage. Data*, 2000, vol. 29. no. 2, pp. 93–104.

[21] H.-P. Kriegel, M. S. Hubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 444–452.

[22] Y. Li and L. P. Maguire, "Selecting critical patterns based on local geometrical and statistical information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1189–1201, Jun. 2011.

[23] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*, vol. 509. Milwaukee, WI, USA: ASQ Quality, 1931.

[24] Z. He, S. Deng, X. Xu, and J. Z. Huang, "A fast greedy algorithm for outlier mining," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2006, pp. 567–576.

[25] T. Idé and H. Kashima, "Eigenspace-based anomaly detection in computer systems," in *Proc. 10th Int. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2004, pp. 440–449.

[26] C. Xia, W. Hsu, M. L. Lee, and B. C. Ooi, "BORDER: Efficient computation of boundary points," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 3, pp. 289–303, Mar. 2006.

[27] Y. Li, "Selecting training points for one-class support vector machines," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1517–1522, 2011.

[28] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "Novelty detection using level set methods," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 576–588, Mar. 2015.

[29] Y. Li, "A surface representation approach for novelty detection," in *Proc. Int. Conf. Inf. Autom. (ICIA)*, Jun. 2008, pp. 1464–1468.

[30] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 55–63.

[31] J. Kubica, A. W. Moore, D. Cohn, and J. G. Schneider, "Finding underlying connections: A fast graph-based method for link analysis and collaboration queries," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 392–399.

[32] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[33] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[34] Y. Fu and T. S. Huang, "Locally linear embedded eigenspace analysis," Univ. Illinois Urbana-Champaign, IFP-TR, Tech. Rep., 2005.

[35] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, Dec. 2003.

[36] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16. 2003, pp. 234–241.

[37] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 663–670.

[38] X. Li, J. C. Lv, and Z. Yi, "Manifold alignment based on sparse local structures of more corresponding pairs," in *Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2013 pp. 2862–2868.

[39] J. Gallier, "Basics of affine geometry," in *Geometric Methods and Applications*. New York, NY, USA: Springer, 2011, pp. 7–63.

[40] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[41] J. Ham, D. D. Lee, and L. K. Saul, "Semisupervised alignment of manifolds," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2005, pp. 120–127.

[42] D. Chen, J. C. Lv, and Z. Yi, "A local non-negative pursuit method for intrinsic manifold structure preservation," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1745–1751.

[43] K. I. Joy. *Affine Combinations, Barycentric Coordinates, and Convex Combinations*, accessed on 2000. [Online]. Available: http://www.idav.ucdavis.edu/education/CAGDNotes/Affine-Barycentric-and-Convex.pdf

[44] D. P. Bertsekas, *Convex Optimization Theory*. Belmont, MA, USA: Athena Scientific, 2009.

[45] J. Gallier, *Geometric Methods and Applications for Computer Science and Engineering*, vol. 38. Berlin, Germany: Springer, 2011.

[46] K. Tang, R. Liu, Z. Su, and J. Zhang, "Structure-constrained low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2167–2179, Dec. 2014.

[47] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[48] S. Chen and D. Donoho, "Basis pursuit," in *Proc. Conf. Rec. 28th Asilomar Conf. Signals, Syst. Comput.*, vol. 1. Oct/Nov. 1994, pp. 41–44.

[49] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[50] F. Korn and S. Muthukrishnan, "Influence sets based on reverse nearest neighbor queries," in *Proc. Int. Conf. Manage. Data ACM SIGMOD*, 2000, vol. 29. no. 2, pp. 201–212.

[51] Y. Tao, D. Papadias, and X. Lian, "Reverse kNN search in arbitrary dimensionality," in *Proc. 30th Int. Conf. Very Large Data Bases*, vol. 30. 2004, pp. 744–755.

[52] C. Xia, W. Hsu, M. L. Lee, and B. C. Ooi, "Border: Efficient computation of boundary points," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 3, pp. 289–303, Mar. 2006.

[53] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 55–63.

[54] L. Fu and E. Medico, "Flame, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinf.*, vol. 8, no. 1, p. 3, Jan. 2007.

[55] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

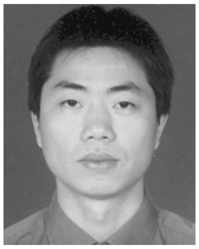
[56] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, p. 4, 2007.

[57] R. Forsyth, *Pc/beagle User's Guide*. London, U.K.: BUPA, 1990.

[58] E. Acuna and C. Rodriguez, "A meta analysis study of outlier detection methods in classification," Dept. Math., Univ. Purto Mayaguez, Mayaguez, Purto Rico, Tech. Rep., 2004.

Xiaojie Li received the Ph.D. degree in computer science and engineering from the College of Computer Science, Sichuan University, Chengdu, China.

She is currently a Lecturer with the College of Computer Science, Chengdu University of Information Technology, Chengdu, China. Her research interests include machine learning, neural networks, and data mining.



Jiancheng Lv (M'09) received the Ph.D. degree in computer science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2006.

He was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He is currently a Professor with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China. His research interests include neural networks, machine learning, and big data.

Zhang Yi (M'08–SM'09–F'16) received the Ph.D. degree in mathematics from the Institute of Mathematics, Chinese Academy of Sciences, Beijing, China, in 1994.

He is currently a Professor with the College of Computer Science, Sichuan University, Chengdu, China. His current research interests include neural networks and big data.