Multidimensional Balance-based Cluster Boundary Detection for High Dimensional Data

Xiaofeng Cao, Baozhi Qiu, Xiangli Li, Zenglin Shi, Guandong Xu, and Jianliang Xu

***Abstract*—The balance of neighborhood space around a central point is an important concept in cluster analysis. It can be used to effectively detect cluster boundary objects. The existing neighborhood analysis methods focus on the distribution of data, i.e., analyzing the characteristic of the neighborhood space from a single perspective, and could not obtain rich data characteristics. In this paper, we analyze the high dimensional neighborhood space from multiple perspectives. By simulating each dimension of a data point's *k* nearest neighbors space (kNNs) as a lever, we apply the lever principle to compute the balance fulcrum of each dimension after proving its inevitability and uniqueness. Then, we model the distance between the projected coordinate of the data point and the balance fulcrum on each dimension, and construct the $DHBlan$ coefficient to measure the balance of the neighborhood space. Based on this theoretical model, we propose a simple yet effective cluster boundary detection algorithm, called Lever. Experiments on both low and high dimensional datasets validate the effectiveness and efficiency of our proposed algorithm.**

***Index Terms*—Cluster boundary, high dimensional space, unlimited lever, balance principle.**

I. Introduction

UNSUPERVISED learning is a process of discovering potentially valuable knowledge that facilitates a better understanding of the underlying data. Of the many methods, cluster analysis [1-5] which aims at learning interesting, nontrivial, and hidden rules from the unknown data, has been widely used in different learning systems such as image segmentation [6-7], biological analysis [8-9], medicine research [10-11], information retrieval [12-13], and natural language processing [14]. A rich set of clustering algorithms has been developed in the literature of neural networks [15-18]. Recently, the deep learning approach has also been applied to clustering [19]. In addition to clustering, cluster boundary detection is another important task of cluster analysis. The boundary data points, with clear class labels, are distributed at the edge of a cluster. They are different from the internal data of a cluster. From the perspective of pattern recognition, cluster boundary points represent the data objects that have a clear ownership but may depart, e.g., people who have been infected by some virus but do not yet suffer from a disease, irregular handwritten characters, target objects which have entered a forbidden area, etc.

To date, researchers have proposed many cluster boundary detection algorithms, such as BORDER [20], BRIM [21], BAND [22], BRINK [23], BERGE [24], and Spinver [25]. These algorithms have gained satisfactory results for low dimensional data based on some geometric theories. However, they are inferior for high dimensional data due to the data sparsity and complexity in high dimensional space. The concept of a cluster boundary was proposed in the DBSCAN algorithm [26]. It randomly selects some core points to search clusters, and the process is terminated when it meets the cluster boundary objects. The distribution of boundary points' nearest neighbors is not uniform. In other words, the neighborhood spaces are not balanced around central points. In contrast, the nearest neighbors of core points are uniformly distributed around the core points. As such, the DBSCAN algorithm can be considered a search process from balance points to imbalance points. It is further observed that many other algorithms also bear the concept of balance, such as *k*-means [27-28], FCM (Fuzzy C-Means) [29-30], and Meanshift [31-34]. The k-means algorithm initializes *k* centroids firstly, and then classifies the data points into *k* clusters. Because the clusters may be imbalanced around the initial centroids, the algorithm iteratively recalculates the centroids of the clusters and reclassifies the data points. The iteration of the algorithm will not stop until all the clusters are balanced around their centroids. The MeanShift algorithm drifts the mean vector by judging whether the module of the mean shift of the current neighborhood is 0. If the module is higher than 0, the neighborhood is not balanced, and the mean vector points to the data points that introduce imbalance. Then, the algorithm continues to update the positions of the central points until the module of the mean shift becomes 0 (i.e., the current neighborhood is balanced).

It is noted that all the aforementioned algorithms are all concerned with neighborhood distributions. In the DBSCAN algorithm, the circular neighborhood around a central point is imbalanced if the number of data points in the neighborhood is less than a preset threshold. The algorithm changes the search direction when it meets the boundary points. In the *k*-means algorithm, a cluster is imbalanced if the central point and the

X. Cao is with the School of Information Engineering, Zhengzhou University, Zhengzhou, China, and the Advanced Analytics Institute, University of Technology Sydney, Sydney, NSW, Australia. Email: xiaofeng.cao@student.uts.edu.au.

B. Qiu and X. Li are with the School of Information Engineering, Zhengzhou University, Zhengzhou, China. E-mail: {iebzqiu, iexlli}@zzu.edu.cn.

Z. Shi is with the University of Amsterdam, Amsterdam, Netherlands. E-mail:zenglin.shi@uva.nl.

G. Xu is with the Advanced Analytics Institute, University of Technology of Sydney, Sydney, NSW, Australia. E-mail: guandong.xu@uts.edu.au.

J. Xu is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR, China. Email: xujl@comp.hkbu.edu.hk.

G. Xu and J. Xu are the co-corresponding authors.

*Data points and objects are used exchangeably in this paper.

*Neighborhood space consists of the local neighbors of one data point and it is a geometric notion.

selected centroid are not the same. In the MeanShift algorithm, the neighborhood is imbalanced if the module of the mean shift is bigger than 0. The concept of balance can be used to distinguish a high-density neighborhood from a low-density one and a uniform neighborhood from a non-uniform one.

To our knowledge, there are three typical neighborhood identification methods, namely, circular range [35], grid range [36], and k nearest neighbors [37-39]. Of these, the circular range will become a hypersphere space and the grid range will become a hypercube space in high dimensional space [40-43]. In order to check the neighborhood distribution, a typical approach is sampling. However, it is challenging to decide the range of the sampling space that has an appropriate number of sample points. If the sampling range is too small, the space may have only a few points and lose important distribution characteristics. On the contrary, if the sampling range is too large, the space may have more than enough points and their distribution characteristics will be hard to analyze. In contrast to the range-based methods, a k-nearest-neighbors (kNN) method always takes the irregular space which is constructed by a point's k nearest neighbors so that the neighborhood distribution can be better captured.

The above sampling methods are focused on the object distribution within the neighborhood, and analyze the data from a single perspective of distance. But the data distribution of high dimensional space is sparse and complex, so judging the balance of neighborhood distribution merely based on the spatial distance of the data may not be accurate. In this paper, we attempt to analyze the data from multiple perspectives, i.e., analyzing the balance of each data dimension to judge the balance of neighborhood space. Regarding the balance, we leverage the classic physics theorem - the lever balance principle. The principle shows a special state of balance when forces are added to the mechanical device. It has been widely used in psychology, physiology, economics, and other fields.

Inspired by the lever balance principle, we take each dimension of a point's kNNs as a separate lever. After proving the inevitability and uniqueness of the balance fulcrum on each dimension, we compute the balance fulcrum of each lever. Then, we construct the $DHBlan$ (i.e., High Dimensional Balance) coefficient to measure the neighborhood balance. Finally, we propose an algorithm, called Lever, to detect cluster boundary objects. The main contributions of this paper are summarized as follows:

- We proposed a novel idea to analyze high dimensional data from multiple perspectives;
- We introduce the lever balance principle to cluster boundary detection and propose to detect the cluster boundary by judging the balance of the neighborhood space;
- We develop the $DHBlan$ coefficient to measure the balance of a high dimensional neighborhood;
- We design a new cluster boundary detection algorithm called Lever for high dimensional data.

The remainder of this paper is organized as follows. Section II introduces the proposed detection model and algorithm, in which the Section II.A describes the physical assumption between cluster boundary and balance of lever, Section II.B reports the used definitions and notations, Section II.C proposes the $DHBlan$ coefficient, and Section II.D proposes the Lever algorithm. To verify the detection ability of the proposed algorithm, Section III.A reports the data preprocessing methods, Section III.B presents the quality evaluation standard, Section III.C to G apply the cluster boundary detection in various scenarios including synthetic, medical, handwritten digit, face images, and target tracking. We then, discuss the parameter settings in Section IV.A, study the scalability of the proposed Lever algorithm with respect to the dataset size and the dimensionality of data objects in Section IV.B, and analyze the importance of $DHBlan$ coefficient in Section IV.C, Our conclusion is given in Section V.

II. Detection Model Based on Lever Balance

The detection model of this paper is inspired by the lever balance principle. Therefore, Section II.A describes the physical assumption between the cluster boundary detection task and balance analysis of lever. Then, Section II.B represents the definition of cluster boundary point and used notations of this paper.

Under this assumption, Section II.C simulates each one-dimension space of a high dimensional neighborhood space as an unlimited lever, in which each data point would be taken as a particle acted by the gravity force. To describe the balance of one dimension around one data point, we propose the $HBlan$ coefficient. By scaling it in the high dimensional space, we then extend it into a high dimensional balance coefficient-$DHBlan$. With the help of this effective detection coefficient, Section II.D proposes the Lever algorithm.

A. Lever and High Dimensional Space

A balance fulcrum is an important physical quantity in the lever balance principle. The position of the balance fulcrum reflects the force bearing status of a lever. If we add a force to the tail of the lever, the balance fulcrum will move toward the tail. If we add a force to a position near the balance fulcrum, the position offset of the balance fulcrum will become smaller. Essentially, the position of the balance fulcrum reflects the force distribution of the lever.

If we regard the forces as data points, the force analysis would become data analysis [44] [45]. When some noise points or isolated points are distributed on the lever, the balance fulcrum will move toward them. Based on this observation, we introduce the lever balance principle to cluster boundary detection for high dimensional data. Specifically, we simulate each dimension of a point's kNNs as a different lever. Then, our analysis can be focused on the levers, rather than the high dimensional space that is abstract and difficult to understand. By computing the distance between the balance fulcrum and the projected coordinate of the point on every dimension (lever), we judge the balance of the neighborhood space. The smaller the distance, the higher the level of balance will be. On the contrary, a bigger distance means more noise points and isolated points that the lever has.

TABLE I: A summary of notations

Variable/Coefficient	Definition/Function
X	Data set with a size of $n \times d$
x_i	A data point or object in X
x_{ij}	The j-th dimension's value of x_i
x_i^l	The l-th nearest neighbor of x_i
x_{ij}^l	the projected coordinate on the j-th dimension of x_i^l
w	A point in the j-th dimension space of x_i
kNN	k nearest neighbors
kNNs	k nearest neighbor space
$\mathcal{H}(\cdot)$	Balance function
F_{ij}^l	The force acting on x_{ij}^l
w^*	The balance fulcrum on the j-th dimension of x_i's kNNs
G	The gravity of a particle
$Blan(x_{ij})$	Measure the balance of the j-th dimension of x_i
$HBlan(x_i)$	Measure the balance of kNNs of x_i
$Diver$	To discretize the $HBlan$ coefficient
$DHBlan$	To detect the cluster boundary points
$a \to b$	The logical operation of a equals b

B. Definitions

Cluster boundary point [20]: A boundary point p is an object that satisfies the following conditions:
1. It is within a dense region IR.
2. $\exists$ region IR' near p, Density(IR')>>Density(IR) or Density(IR')<<Density(IR).

Notations: Table I lists the variables (rows 1-9) and coefficient functions (rows 10-13) used in this paper.

C. Unlimited lever

According to the above analysis, each dimension of x_i's kNN is simulated as a different lever and each data point is simulated as a particle acted by the gravity force. Given a variable fulcrum w on the j-th dimension space, we construct a balance function $\mathcal{H}$ to represent the balance relation between w and the j-dimension coordinates of x_i's kNNs. It is formally defined as follows:

$$\mathcal{H}(w) = \delta_w \gamma_w^T \tag{1}$$

where $\gamma_w = (x_{ij}^1 - w, x_{ij}^2 - w, ..., x_{ij}^l - w)$, $\delta_w = (F_{ij}^1, F_{ij}^2, ..., F_{ij}^l)$, x_{ij}^l $(l = 1, 2, ..., k)$ is the projected coordinate on the j-th dimension of object x_i's l-th nearest neighbor, and $F_{i,j}^l$ is the force that acts on x_{ij}^l. If the lever has a balance fulcrum w^*, it must satisfy the condition:

$$\mathcal{H}(w^*) \to 0 \tag{2}$$

For an unlimited lever, variable $w \in (-\infty, +\infty)$ and Eq. (1) has a linear relationship with w. So, there must be a point that can make $\mathcal{H}(w^*) \to 0$. Then, Eq. (2) will be true. To prove the inevitability and uniqueness of the balance fulcrum in the j-dimension space, we will show that the first-order partial derivative of $\mathcal{H}$ is monotonic.

According the above analysis, our proof goal is:

$$\exists w \in (-\infty, +\infty), \mathcal{H}(w^*) \to 0 \tag{3}$$

The formal proof is shown as follows.

Proof.

$$\frac{\partial \mathcal{H}(w)}{\partial w} = -\delta_w I^T \tag{4}$$

where $I = (1, 1, ..., 1)_{1\times k}$. Intuitively, because $\frac{\partial \mathcal{H}(w)}{\partial w} < 0$, $\mathcal{H}(w)$ is a monotonically decreasing function. When w increases, $\mathcal{H}(w)$ will decrease. In the real lever system, due to the length limitation of the lever, the lever fulcrum may not exist. But in the data space, we simulate the dimension as an unlimited lever, i.e., $w \in (-\infty, +\infty)$. Thus, we can obtain the following results:

$$\begin{cases} \mathcal{H}(w) \to \epsilon_1, \epsilon_1 < 0, & if\ w > x_{ij}^l, \forall l \in (1, k) \\ \mathcal{H}(w) \to \epsilon_2, \epsilon_2 > 0, & if\ w < x_{ij}^l, \forall l \in (1, k) \end{cases} \tag{5}$$

where ϵ_1 and ϵ_2 are constants. Because $\mathcal{H}(w)$ is monotonically decreasing, when $w \in (min(x_{ij}^l), max(x_{ij}^l)), \forall l \in (1, k)$, there must exist a unique balance fulcrum. In other words, w^* uniquely exists.

Now, we solve Eq. (2):

$$\delta_w \gamma_w^T = \delta_w \mathcal{X}^T - \delta_w I^T w^* \to 0 \tag{6}$$

where $\mathcal{X} = (x_{ij}^1, x_{ij}^2, ..., x_{ij}^l)$, then

$$w^* = \frac{\delta_w \mathcal{X}^T}{\delta_w I^T} \tag{7}$$

For the lever constructed for each dimension, each data point has the same quality and $F_{ij}^l \to G$, where G is the gravity of a particle. Hence, Eq. (7) can be rewritten as:

$$w^* = \frac{\mathcal{X} I^T}{I I^T} \tag{8}$$

Proof end.

To measure the balance of the dimension, we need to measure the similarity between $\mathcal{H}(w)$ and $\mathcal{H}(w^*)$, so:

$$\begin{aligned} \epsilon_3 &= |\mathcal{H}(w) - \mathcal{H}(w^*)| \\ &= |\gamma_w \delta_w^T - \gamma_{w^*} \delta_w^T| \\ &= |(\gamma_w - \gamma_{w^*}) \delta_w^T| \\ &= \tau_1 \Lambda I^T \\ &= \tau_2 |w - w^*| \end{aligned} \tag{9}$$

where $\Lambda = (|w^* - w|, |w^* - w|, ..., |w^* - w|)_{1\times k}$, ϵ_3, τ_1, τ_2 are three constants. So, we propose a $Blan$ coefficient to measure the balance of each dimension, and its definition is:

$$Blan(x_{ij}) = |x_{ij} - w^*| \tag{10}$$

If the $Blan$ coefficient is 0, the lever system will be balanced. As for the data points, this means that the neighbors are distributed uniformly around the central point (i.e., x_{ij}). The farther the distance between the balance fulcrum and the

central point, the bigger the value of $Blan$ coefficient, and the less uniform the data distribution and the lever system would be. Note that the $Blan$ coefficient only reflects the balance of the lever system for the j-th dimension. It cannot be used to measure the balance of all the lever systems, i.e., the high dimensional neighborhood space consisting of many different dimensions. We assume that the weight of each dimension is the same. Consequently, we can use the sum of the $Blan$ coefficient of each lever to measure the balance of x_i's kNNs, and propose the $HBlan$ coefficient:

$$HBlan(x_i) = \Xi I^T \tag{11}$$

where $\Xi = (|x_{i1} - w_1^*|, |x_{i2} - w_2^*|, ..., |x_{im} - w_m^*|)$, and w_j^* is the best balance fulcrum in the j-dimension of x_i's kNNs.

There are three types of data objects in a dataset: noises, cluster boundary objects, and core objects. The task of cluster boundary detection aims to classify the three types of data objects. An efficient method should be able to quickly capture the unique characteristics of boundary objects. We propose to use the $HBlan$ coefficient to detect the cluster boundary. Because the kNNs of a core object is uniform, the balance of the space of core objects is strong and the $HBlan$ coefficient values are small. On the other hand, the $HBlan$ coefficient values of cluster boundary objects are generally larger than those of core objects, while noises are expected to have the largest $HBlan$ coefficient values.

However, in the real world, some datasets may have a lot of noises and the kNNs of some noises may be sparse. As a result, their $HBlan$ coefficient values may be close to those of boundary objects. To reduce the influence of such noises, we use the $divergence$ of kNN to discretize the $HBlan$ coefficient. The $divergence$ is defined as follows:

$$Diver(x_i) = \Theta I^T \tag{12}$$

where $\Theta = (e^{||x_i^1 - x_i||_2}, e^{||x_i^2 - x_i||_2}, ..., e^{||x_i^k - x_i||_2})$ and x_i^j is the j-th nearest neighbor of x_i. Due to the sparsity of noises and isolated points, their Diver values would be relatively large. Finally, we propose the $DHBlan$ coefficient as the product of $Diver$ and $HBlan$:

$$DHBlan(x_i) = \Theta I^T \Xi I^T \tag{13}$$

If the $DHBlan$ coefficient of a data object is relatively large, the data object may be a noise. Otherwise, it may be a core object. Therefore, we can get the following inequality:

$$DHBlan(noise) > DHBlan(boundary) > DHBlan(core) \tag{14}$$

Then, we describe Eq. (13) in a probability density function (PDF) to further show Eq. (14):

$$f(x_i) = 1 - \frac{1}{\sqrt{2\pi}} exp(-\Theta I^T \Xi I^T) \tag{15}$$

We plot the curve of this function in Figure 1, which shows the PDF change on the normalized data (the normalization step is to be detailed in Algorithm 1). Then, we can use the $DHBlan$ coefficient to detect the cluster boundary objects and the proposed Lever algorithm will be introduced in the next section.

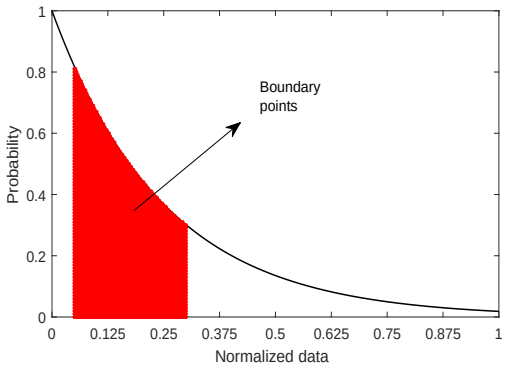


Fig. 1: The PDF change on the normalized data.The x-axis of colored block represents the boundary points while the noises and core points lie to the left and the right, respectively.

D. Lever algorithm

Based on the proposed $DHBlan$ coefficient, we now propose a cluster boundary detection algorithm, named Lever. Firstly, we find the kNN objects for each data object (line 6). Then, we compute their $DHBlan$ coefficient values and store them in an array (lines 7-9). Next, we get the normalized serial number of each data object (lines 11-18). Finally, we identify the cluster boundary objects according to the input parameters (lines 19-23). In addition, we suggest users normalizing the data sets with large value ranges into the range [0, 1] before running the algorithm, such as the data preprocessing methods used in the experiment section. One reason is to reduce memory consumption and the other is to balance the two parts of our objective function.

III. EXPERIMENTS

In this section, we conduct a series of experiments to evaluate the performance of the proposed Lever algorithm:

- Sections III.A reports the used data preprocessing methods of this paper;
- Sections III.B describes the quality evaluation method of the cluster boundary detection task;
- Sections III.C and III.D compare the boundary detection ability of different algorithms on some synthetic and medical datasets;
- Sections III.E and III.F further examine Lever's boundary detection ability on handwritten digits and face image datasets;
- Section III.G details some interesting experiments on target tracking.

A. Data preprocessing

The datasets used are summarized in Table II. Before conducting the experiments, we perform some preprocessing on these datasets. The preprocessing methods are as follows:

Algorithm 1. Lever

Input: X // data set; k // number of nearest neighbors;
λ_1 // lower bound; λ_2// upper bound
Output: $\mathcal{Y}$ // boundary object set
Initialization: $i, j \leftarrow 1$; $\mathcal{A}, \mathcal{B}, \mathcal{R}, \mathcal{Y} \leftarrow \varnothing$
Begin:
Find the kNN set of each data object in X;
  for i=1 to n **do**
  $\mathcal{A}(i)$ = $DHBlan(x_i)$
  endfor
Sort $\mathcal{A}$ by descending order and store them in matrix $\mathcal{B}$
  for i=1 to n **do**
    for j=1 to n **do**
      if $\mathcal{A}(i) - \mathcal{B}(j) \rightarrow 0$ **do**
      $\mathcal{R}(i) \leftarrow j/n$ // calculate the sort number and normalize them
       break
      endif
    endfor
  endfor
  for i=1 to n **do**
    if $\mathcal{R}(i) \in (\lambda_1, \lambda_2)$ **do** // detect the cluster boundary
    $\mathcal{Y} \leftarrow \mathcal{Y} \cup x_i$
    endif
  endfor

TABLE II: Datasets and preprocessing methods

D	n	Dimensionality	Preprocessing
Mnist	10,000	28	(c)
Colon	62	2,000	(a)
Prostate	102	10,509	(b)
Pointing	1,395	384	(c)
Waving Trees	287	160	(c)
Moved Object	1,745	160	(c)

- the value of each dimension of each data object is divided by 10^3 ;
- the value of each dimension of each data object is divided by 10^4;
- for each image, read the $x \times y$ grayscale matrix and compress it into a single-column matrix (i.e., with a size of $1 \times y$) with the average grayscale values.

Preprocessing steps (a) and (b) aim to reduce the computation overhead (speed and memory) since the data domain is large. Then the values will be normalized to [0, 1] after being proprocessed. Step (c) is an image processing approach which transforms the image type to numerical type.

B. Quality evaluation

In cluster analysis, there are two methods that can be used for quality evaluation [46-47]. The first is a supervised method based on trained data classification results. It requires knowing the class label of each data object before the data is analyzed. The second is unsupervised and takes the separability and compactness of clusters as the evaluation standard.

In our study, we focus on the cluster boundary of all the clusters, not on that of each single cluster. The cluster boundary detection results are unique. We only need to analyze whether each data object is a boundary object or not. In this paper, we use the accuracy rate and the recall rate to evaluate the effectiveness of the detection results, and adopt the F-measure [48-49] as a comprehensive performance evaluation metric. The related definitions are as follows:

$$Precision = \frac{a}{b}$$
$$Recall = \frac{a}{d}$$
$$F\text{-}measure = \frac{2}{\frac{a}{b} + \frac{a}{d}}$$

where a is the number of real boundary objects detected, b is the number of detection results, and d is the number of real boundary objects. The accuracy rate and the recall rate complement each other. When the algorithm detects most of the real boundary objects (i.e., achieves a high recall rate), we cannot immediately say that the algorithm is good. In cases where the detection results also include a lot of noises or core points, the detection results may suffer from a low accuracy rate. The F-measure combines these two metrics and serves as an overall performance metric.

C. Synthetic datasets

Fig. 2 shows four different synthetic data sets, namely DS1, DS2, DS3 and DS4 [25]. There are 7,832, 5,034, 5,400, and 4,800 data points in these four data sets, among which 640, 538, 1,077, and 1,204 are cluster boundary points, respectively. DS1 contains two diamond clusters, and the two clusters are close. DS2 contains five clusters surrounded by a lot of noises. DS3 contains three elliptic clusters and noises are located near the edge of clusters. DS4 includes a circular cluster and an annulus cluster, and the noises are distributed uniformly between the clusters. Fig. 3 shows the best boundary detection results of different algorithms on these four datasets. Detailed experimental results are reported in Table III.

In the cluster boundary detection results of BORDER(see Figs. 2(a) and 3(a)), as the number of noises' reverse kNNs is less than that of boundary points, all the noises and isolated points are detected as boundary points by mistake. The BAND and BRINK algorithms use the variable coefficient to detect the cluster boundary and get a better performance than BORDER. As shown in the detection results of BAND(see Figs. 2(b) and 3(b)), the noises and isolated points located far away from clusters are filtered precisely, but the noises near the cluster edges are still detected as cluster boundary points. Similar results are observed for BRINK. The reason for this is that these special noises have similar neighborhood distributions with the boundary points. The BERGE algorithm uses the idea of statistical learning to detect the cluster boundary objects. But a wrongly labeled result will affect its subsequent labeling process. Thus, it cannot be successfully used in the datasets with a lot of noises. The Spinver algorithm applies the theory of space inversion to convert the static space into a dynamic space. It uses the improved Hopkins statistics to judge the uniformity of the neighborhood space. A 2-d Gaussian filter is also employed to smooth the noises.

The results reported in Table III show that Spinver performs better than BORDER, BAND, and BRINK, but still worse than Lever. In the detection results of Lever(see Figs. 2(c) and 3(c)), all the noises are filtered accurately; its boundary detection results are more accurate than those of other algorithms. This verifies the effectiveness of our proposed model that leverages the multidimensional balance of the neighborhood space for cluster boundary detection.

D. Medical datasets

Cancer prevention and treatment are challenging issues in medical research. Because of the long incubation period of malignant tumor viruses and no obvious symptoms in the early stage of illness, cancers are difficult to discover until they evolve to the terminal stage. In medical databases, clustering can classify people as normal and patients. We often focus on patients, but ignore the abnormal individuals of normal people. These individuals may have been infected by the virus but have not yet suffered from the disease. The effective detection of these people not only ensures they receive prophylactic treatment, it also enables the incubation period characteristics of cancers to be studied. Here, we define such individuals as the cluster boundary points of normal people. Similarly, the cluster boundary may be defined as the objects which carry the recessive infection virus or mutant genes. Our work may help medical researchers in further research.

The Biomed dataset [50] has 134 normal objects and 75 objects which have been infected by the virus. Of the normal objects, there are 30 virus carriers, who are defined as the cluster boundary of normal people. The Cancer dataset [51] has 241 malignant tumor objects and 75 benign tumor objects. Of these, 30 benign tumor objects may become malignant tumor patients, and they are considered as the cluster boundary of normal people. The Colon dataset [52] is a colon cancer gene expression [53] dataset with 62 samples, including 22 normal samples and 40 colon cancer samples. Each sample has 2,000 genes. The Prostate dataset [54] is also a gene dataset, which has 102 samples, including 50 norm samples and 52 prostate cancer samples. In this dataset, each sample has 10,509 genes. Before the experiments, we perform statistical experiments on DBSCAN to get seven cluster boundary objects for the Colon dataset and 18 cluster boundary objects for the Prostate dataset. Then, we preprocess these datasets according to Table II.

As shown in Table III, BAND has the worst cluster boundary detection performance. While the BRINK algorithm uses the weighted Euclidean distance to measure the similarities between data objects, its performance is better than BAND. Regarding BORDER, although it cannot separate the noises and isolated points, its detection results include most of the real boundary points. Since the datasets tested here are small and have no or few noises, the drawback of BORDER is not apparent and a good performance is gained. BERGE uses the idea of evidence accumulation to detect the cluster boundary. But the algorithm is sensitive to the centers of clusters. When the dataset has a small number of samples, this algorithm cannot obtain credible results. Spinver applies the Hopkins statistics to detect the cluster boundary objects. Its main drawback is that it cannot separate the noises and boundary points accurately. But because the datasets tested have few noises, the detection results are good, being only slightly worse than that of Lever. Clearly, the best performance is achieved by the Lever algorithm, which simulates the high dimensional space as many levers and uses the $DHBlan$ coefficient to detect the boundary objects. The results validate the effectiveness of the proposed $DHBlan$ coefficient.

E. Handwritten digits

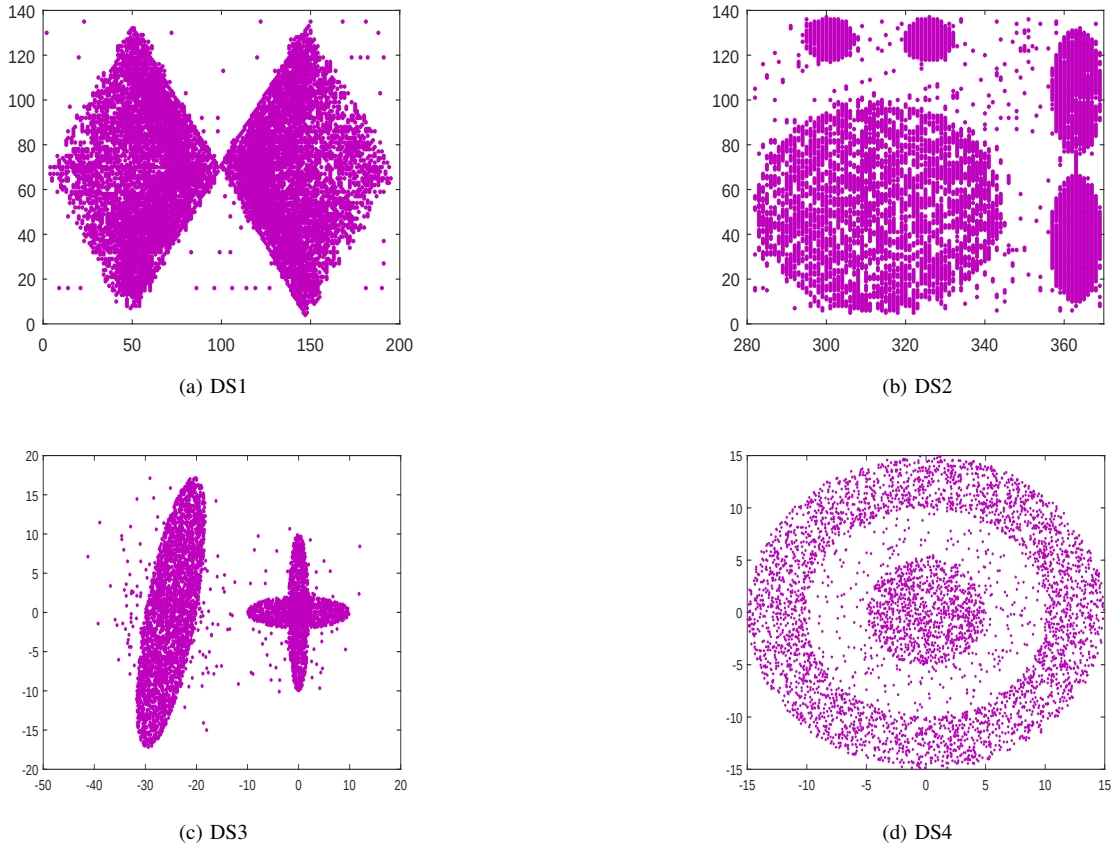
Next, we perform some experiments on image datasets to further verify the performance of Lever. In the fields of identity authentication, code scanning, signature recognition, and handwritten digit recognition [55-56] are of important value and practical significance. Due to personal preferences and habits, there are big differences in digital shapes, sizes, and line widths for the same digit. The cluster boundary is thus defined as the digit images which appear difficult to recognize.

The Mnist dataset [57] contains 10 handwritten digits, including 60,000 training image samples and 10,000 test image samples. The images are stored in 8-bit depth BMP formats. Each image has 28×28 pixels, and each pixel has a gray value in the range of 0-255. We choose the handwritten digit '8' (974 images) from the image samples to detect the cluster boundary using the Lever algorithm and the result is presented in the Fig. 4. It can be seen that Lever effectively detects the irregular images as cluster boundary objects and the relatively standard digits as cluster core objects. This demonstrates the effectiveness of the Lever algorithm for cluster boundary detection in high dimensional space.

F. Face images

With the increasing abundance of computer image theories [58] and the support of machine learning [59] and deep learning [60], face recognition techniques have developed rapidly [61]. In these applications, the facial features of humans are used to match faces. Compared to normal face images, boundary face objects are those images that have features of strong illumination, faint illumination, sunglasses, or face profiles. As such images affect the accuracy of face recognition, effectively detecting them provides an important reference for face image feature extraction and face recognition.

The Pointing dataset [62] contains different head posture images of 15 volunteers. Each volunteer has 9 postures in the vertical direction and 13 postures in the horizontal direction. Pointing includes two sequences and we take the first sequence in our experiments. The first sequence has a total of 1,395 images, with 93 images from each volunteer. The image format is JPG with 8-bit depth. The pixel size is 288×384 and the grey level of each pixel ranges from 0 to 255. Before conducting the experiments, we transformed these images into a $1{,}395\times384$ matrix. We choose 93 face images of a volunteer to detect the cluster boundary objects (see Fig. 5). The results for all face images are shown in Fig. 6. As can be seen, the detected images all have a large side angle on horizontal and/or vertical directions, suggesting the success of detecting boundary face objects.



(a) DS1

(b) DS2

(c) DS3

(d) DS4

Fig. 2: Synthetic datasets.

G. Target tracking

Target tracking [63-65] is a complex research area in computer vision. Researchers leverage image segmentation techniques to capture moving target objects. Usually, target objects are dynamically moving, but the background environment of the video can be static or dynamic. Thus, the target tracking research can be divided to two categories, i.e., target tracking under a static scene and target tracking under a dynamic scene. Here, different from computer vision researchers, we attempt to use our Lever algorithm to detect moving targets. The Waving Trees [66] is a public dataset released by Microsoft Research. It contains data on the continuous monitoring of the scene of one building. In the process of monitoring, a volunteer passes by the monitored area. In the captured video, no other volunteers or animals broke into the monitored area, and only the trees are waving in the wind. The dataset has 287 images, eight images of which capture the volunteer. Thus, our objective is to detect the eight volunteer images. The Moved Object dataset is another public dataset from Microsoft Research. It contains data on the continuous monitoring of a scene of an office. A volunteer walks into the office and leaves after a period of time. The dataset has 1,745 images, 363 of which are volunteer images to be detected.

Fig. 7 shows the cluster boundary detection results for the Waving Trees dataset. As there are a total of 363 cluster boundary objects for the Moved Object dataset, it is too busy to show all of them. We show 50 of these objects in Fig. 8. The detailed detection results of the different algorithms are reported in Table III.

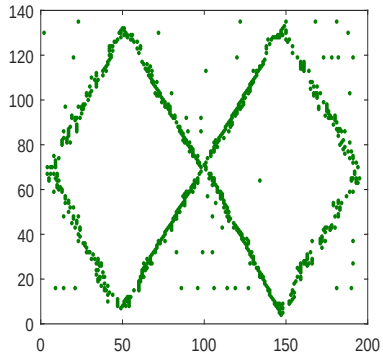
From the experimental results, it is clear that the performances of BAND and BRINK are similar. It was observed in the experiments that it is hard for these two algorithms to detect a special boundary image (see Fig. 9(a)). Because the image only includes a small part of the volunteer's body, it is very similar to the images which do not capture the volunteer. The performances of BORDER and Spinver are good for the Waving Trees dataset, but they are not so good for the Moved Object dataset. The main reason is that the latter dataset has some special images too (see Fig. 9(c)). Actually, before the volunteer enters the office, the monitor captures 1,138 static images (see Fig. 9(b)). After the volunteer leaves, the chair's position has slightly changed and the monitor captures 244 images. These 244 images have an influence on the performance of BORDER and Spinver. On the whole, Lever outperforms all the other algorithms.

IV. DISCUSSIONS

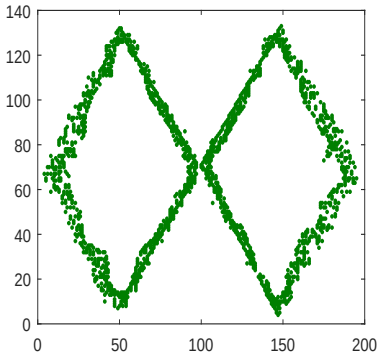
To further analyze the properties of the Lever algorithm, Section III.A discusses the functions of the input parameters of the Lever algorithm and gives reasonable parameter suggestions after a series of tests. Then, Section III.B studies the scalability of the proposed Lever algorithm with respect

TABLE III: The boundary detection results of different algorithms on different data sets.

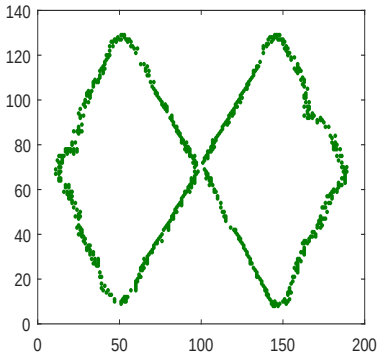
Data sets	Algorithms	Dimensions	Real points on boundary	Points detected	Correct points	Precision rate	Recall rate	F-measure
DS1	BAND	2	640	823	556	0.6756	0.8688	0.7601
	BORDER			723	540	0.7469	0.8438	0.7924
	BRINK			667	520	0.7795	0.8125	0.7957
	BERGE			662	532	0.8036	0.8313	0.8172
	Spinver			611	542	0.8871	0.8469	0.8665
	Lever			620	576	0.9290	0.9000	0.9143
DS2	BAND	2	538	749	454	0.6061	0.8439	0.7055
	BORDER			669	445	0.6366	0.8271	0.7195
	BRINK			499	438	0.8778	0.8141	0.8447
	BERGE			553	472	0.8535	0.8773	0.8652
	Spinver			540	482	0.8926	0.8959	0.8942
	Lever			540	503	0.9315	0.9349	0.9332
DS3	BAND	2	1077	1629	961	0.5899	0.8923	0.7103
	BORDER			1252	831	0.6637	0.7716	0.7136
	BRINK			1540	914	0.8935	0.8478	0.6985
	BRIM			1880	935	0.7870	0.8682	0.8256
	Spinver			1049	993	0.9466	0.9220	0.9341
	Lever			1032	1002	0.9709	0.9304	0.9502
DS4	BAND	2	1204	1944	1056	0.5432	0.8771	0.6709
	BORDER			1802	1089	0.6043	0.9045	0.7246
	BRINK			1817	1003	0.5520	0.8331	0.6640
	BRIM			1355	1062	0.7838	0.8821	0.8300
	Spinver			1264	1111	0.8790	0.9228	0.9003
	Lever			1205	1108	0.9195	0.9203	0.9199
Biomed	BAND	4 30	26	22	0.8462	0.7333	0.7857	
	BORDER			26	23	0.8846	0.7667	0.8214
	BRINK			36	30	0.8333	1.0000	0.9089
	BERGE			26	24	0.9231	0.8000	0.8572
	Spinver			29	27	0.9310	0.9000	0.9153
	Lever			29	27	0.9310	0.9000	0.9153
Cancer	BAND	10	37	37	25	0.6757	0.6757	0.6757
	BORDER			37	28	0.7568	0.7568	0.7568
	BRINK			37	29	0.7837	0.7837	0.7837
	BERGE			37	28	0.7568	0.7568	0.7568
	Spinver			35	34	0.9714	0.9189	0.9444
	Lever			34	34	1.0000	0.9189	0.9577
Colon	BAND	2000	7	6	5	0.8333	0.7143	0.7692
	BORDER			7	7	1.0000	1.0000	1.0000
	BRINK			6	5	0.8333	0.7143	0.7692
	BERGE			6	5	0.8333	0.7143	0.7692
	Spinver			7	7	1.0000	1.0000	1.0000
	Lever			7	7	1.0000	1.0000	1.0000
Prostate	BAND	10,509	18	17	16	0.9412	0.8889	0.9143
	BORDE			19	18	0.9474	1.0000	0.9730
	BRINK			17	16	0.9412	0.8889	0.9143
	BERGE			17	16	0.9412	0.8889	0.9143
	Spinver			18	18	1.0000	1.0000	1.0000
	Lever			18	18	1.0000	1.0000	1.0000
Waving Trees	BAND	160	17	17	15	0.8824	0.8824	0.8824
	BORDE			17	17	1.0000	1.0000	1.0000
	BRINK			17	15	0.8824	0.8824	0.8824
	BERGE			17	15	0.8824	0.8824	0.8824
	Spinver			17	17	1.0000	1.0000	1.0000
	Lever			17	17	1.0000	1.0000	1.0000
Moved Object	BAND	160	363	250	250	1.0000	0.6887	0.8157
	BORDE			363	222	0.6116	0.6116	0.6116
	BRINK			250	244	0.9760	0.6722	0.7961
	BERGE			363	250	0.6887	0.6887	0.6887
	Spinver			363	222	0.6116	0.6116	0.6116
	Lever			363	356	0.9807	0.9807	0.9807



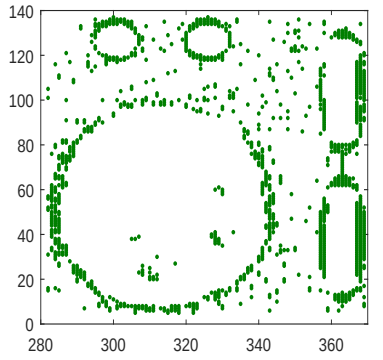
(a) BORDER (k=50, n=500)



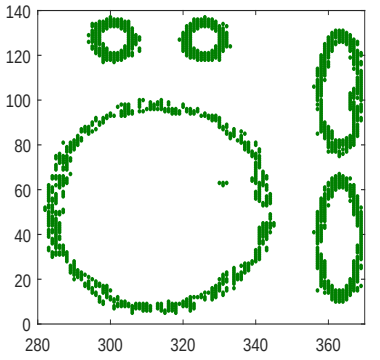
(b) BAND (k=50, w=0.65, BPT=0.66)



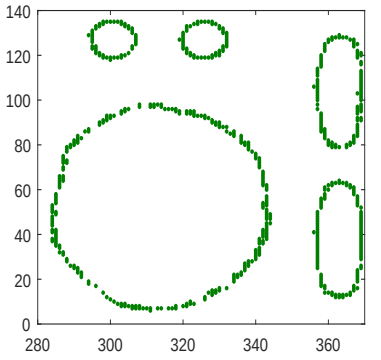
(c) Lever (k=100, λ_1 =0.0391, λ_2 =0.1251)



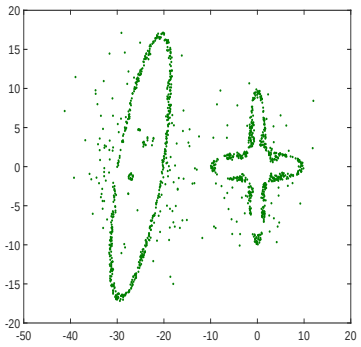
(d) BORDER (k=120, n=1200)



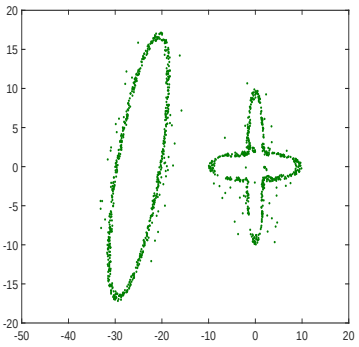
(e) BAND (k=50, w=0.40, BPT=0.80)



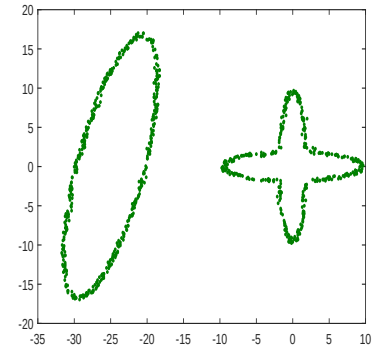
(f) Lever (k=120, λ_1=0.1041, λ_2=0.2239)



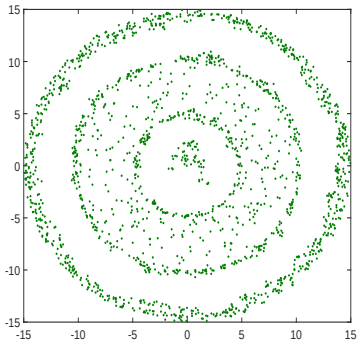
(g) BORDER(k=100,n=1252)



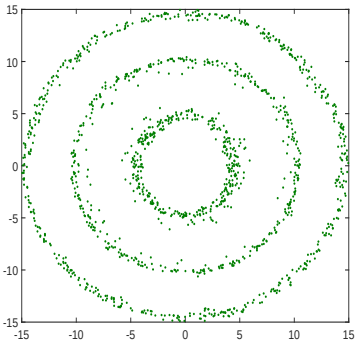
(h) BRIM(Eps=4,δ=1146.9)



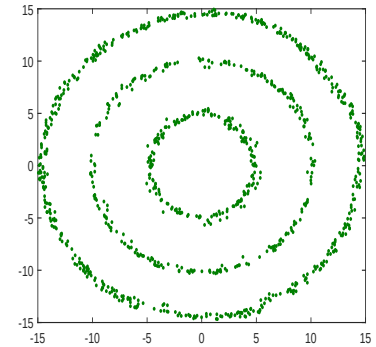
(i) Lever (k=70, λ_1=0.0411, λ_2=0.1326)



(j) BORDER(k=100,n=1802)

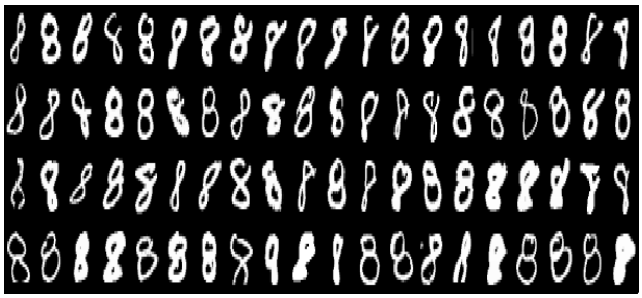


(k) BRIM(Eps=2, δ=80.75)

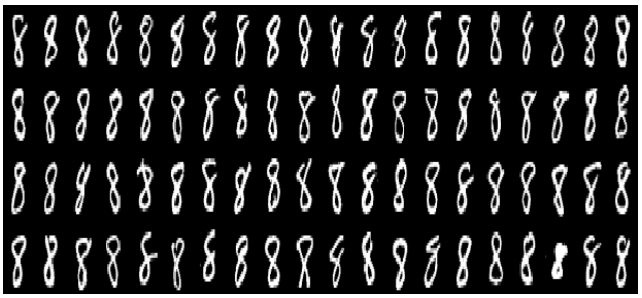


(l) Lever(k=70, λ_1=0.0312, λ_2=0.1407)

Fig. 3: The best detection results of different algorithms on synthetic datasets.



(a) cluster boundary objects



(b) core objects

Fig. 4: The cluster boundary and core objects of '8'.

Fig. 5: The boundary detection result of Lever on a volunteer.

Fig. 6: The boundary detection result of Lever on the first sequence of Pointing dataset.

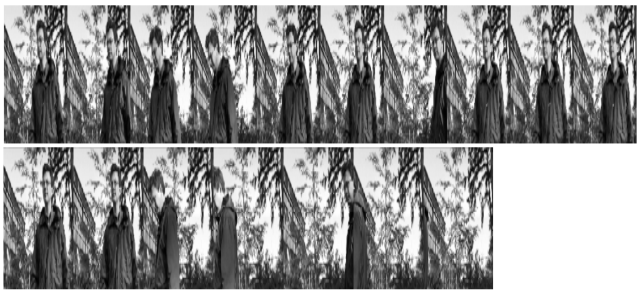
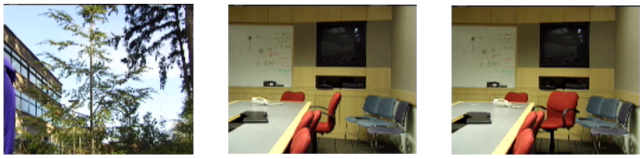


Fig. 7: The boundary detection result of Lever on Waving Trees.

to the dataset size and the dimensionality of data objects. Considering the Lever algorithm is based on the $DHBlan$ coefficient, we present some interesting discussions involved with its advantages in Section III.C.

Fig. 8: The part of boundary detection result of Lever on Moved Object.



(a) (b) (c)

Fig. 9: Special images.

A. Parameter settings

The Lever algorithm uses three parameters to detect the cluster boundary. A lot of experiments show that when $k \in [10, 100]$, $\lambda_1 \in [0.02, 0.05]$, $\lambda_2 \in [0.15, 0.30]$, the algorithm achieves good performance. and reflect the number of core objects, boundary objects, and noises. It is observed that noises usually occupy 2%-5% of the whole dataset, boundary objects are 13%-25%, and core objects are 70%-85%.

To obtain the optimal parameter settings, we perform experiments on the datasets used in the paper. The results are reported in Figs. 10-12 to show the change of F-measure when setting different k, λ_1, λ_2 values, respectively. In these figures, Digit '8' refers to the dataset of handwritten digit '8', Waving refers to the Waving Trees dataset, and Moved refers to the Moved Object dataset. It is important to note that $\lambda_1 = 0$ in the datasets without noises, and we only use λ_2 to detect the boundary for those datasets. Hence, the experiment shown in Fig. 11 does not include the datasets without noises. It is found that when the value of k is located in the range [10, 100], the algorithm can achieve good boundary detection results.

Compared to the dataset size, the value of k is relatively small. In practice, the dataset size has an influence on the selection of k, which cannot be too big or small. λ_1 controls the number of noises. The bigger it is, the more noises that are filtered. But some of these filtered data objects may be real boundary objects. For this reason, with an increase of λ_1, the algorithm filters more boundary objects and the F-measure values drop too. When λ_1 is fixed, we can still use λ_2 to achieve good detection accuracy. λ_2 is used to separate the boundary and core objects. The smaller it is, the more real boundary objects will be lost. The bigger it is, the more core objects will be detected as boundary objects. Consequently, when λ_2 is too small or too large, the F-measure is small. Figs. 10-12 show when $k = 30, \lambda_1 = 0.02, \lambda_2 = 0.2$, the algorithm obtains the best results. As such, it is advised to use these settings for cluster boundary detection.

Nonparametric cluster boundary detection remains a challenge. In fact, cluster boundary detection is to choose one type of objects from all the objects. Capturing the characteristics which are different from other types of data objects is essential to detect the boundary objects. Separating the objects without using any parameter is challenging because some data objects share similar characteristics and are hard to distinguish.

B. Scalability

Theory analysis shows that the time complexity of all the algorithms evaluated are $O(n^2)$ [25]. A set of experiments is conducted to compare their runtime performance.

The results are reported in Fig. 13, where the dataset size is varied from 2,000 to 20,000, and Fig. 14, where the data dimensionality is varied from 500 to 10,000. It can be seen that when the runtimes of BAND, BRINK, BERGE, and Spinver are close to each other, Lever achieves the best performance. The BORDER algorithm uses the reverse kNN to detect the cluster boundary. Because much more time is consumed in the kNN computation, it has the worst performance. The main time consumption of BAND is the computation of the coefficient variation. Compared to BAND, BRINK needs to compute the weighted Euclidean distance in addition to the coefficient variation. Hence, BRINK costs more time than BAND. The BERGE algorithm needs to label the cluster boundary objects many times; so its runtime quickly increases with dataset size. The main time consumption of Lever is the kNN computation, and the time complexity of this process is $O(n^2)$. Also, computing the $DHBlan$ coefficient costs some time. But the time complexity of this step is only $O(n)$. Therefore, reducing the time consumption of kNN computation is the key for Lever to perform better than all the other algorithms.

C. DHBlan coefficient

Our proposed method uses the $DHBlan$ coefficient to detect the cluster boundary and the experiments show promising results. The biggest advantage of the proposed algorithm is the effective separation of the noises from the datasets. This section discusses this interesting aspect. Detecting noises is known as noise detection in data mining research and noise smoothing in image analysis. In data mining and pattern recognition, a cluster is defined as a pattern. Each cluster has a special and different distribution, density, and structure. Generally, the data objects are categorized into two types, i.e., objects within the cluster and objects outside the cluster. The second type of objects are noises. But some noises located far from the clusters may also have a high density. Thus, the concept of isolated objects has been proposed to analyze such special data objects. With the study of cluster boundary detection, it is observed that the data objects located at the edge of a cluster also have important value. As such, the objects within a cluster are further classified into two types, i.e., core objects and boundary objects.

How to separate the noises from a dataset is a challenging problem. In data mining, researchers use the density [67-68] or distribution [69-71] to eliminate noises or isolated points. Furthermore, clustering [72-73] and classification algorithms [74-75] provide functions to eliminate them. In image analysis, the pixel distribution characteristic is used to smooth noises. Also, advanced techniques such as Fourier transform and wavelet transform [76-77] have been proposed. In our proposed method, we take the viewpoint of ''object separation" to smooth noises. Identifying the key differences between different data objects helps us recognize noises. Noises are always located far from clusters, and the most significant feature is the low density. The balance of their neighborhood space is very weak. Regarding cluster boundary objects, most of their neighbors are core objects and some others are noises. Hence, its density is higher than that of noises, but lower than that of core objects. The balance of their neighborhood space is stronger than that of noises. Core objects are distributed in high-density areas and their neighbors are uniformly distributed on each dimension. The balance of their neighborhood space is the strongest. As such, we use Eq. (11) and Eq. (12) to describe the balance and diversity of the neighborhood space, respectively. Finally, Eq. (13), which integrates Eq. (11) and Eq. (12), is employed to detect the boundary objects. In a nutshell, our proposed method can be summarized as *objects separation*. Different from traditional methods, we analyze the data distribution of each dimension to judge the balance of the neighborhood space.

V. CONCLUSION

The balance of neighborhood space around a central point is an important concept and has interesting applications in data mining. Existing methods for identifying the balance of neighborhood space, based on single-perspective analysis, all focus on the neighborhood distribution characteristics of objects. Due to the characteristics and sparsity of high dimensional space, single-perspective analysis cannot obtain much valuable information. In this paper, we proposed the idea of analyzing high dimensional space from multiple perspectives, i.e., multidimensional balance. By simulating the high dimensional space as levers, we proved the inevitability and uniqueness of the existence of the balance fulcrum. We applied the lever balance principle to solve the cluster boundary detection problem in high dimensional space. Experiments based on

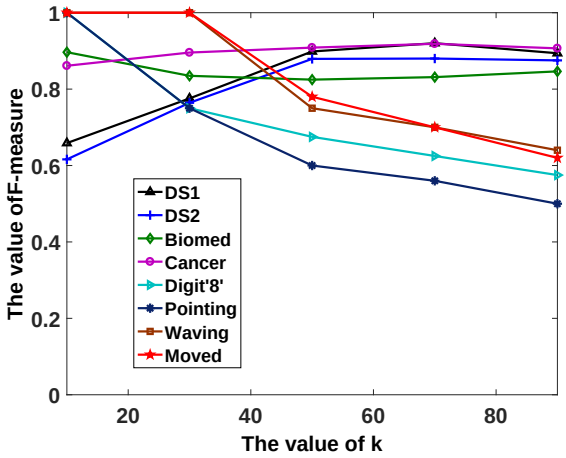


Fig. 10: The change of F-measure when setting different k values on different datasets.

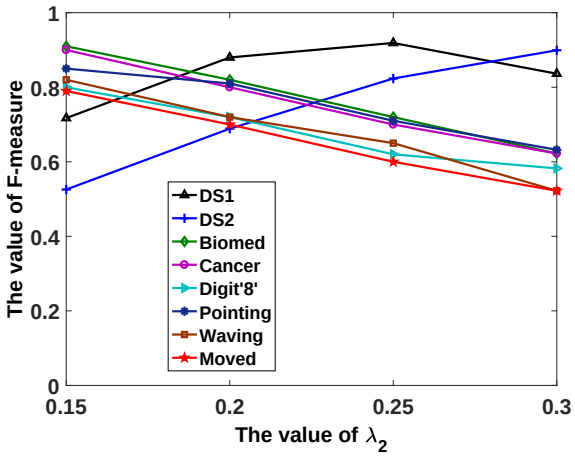


Fig. 12: The change of F-measure when setting different λ_2 values on different datasets.

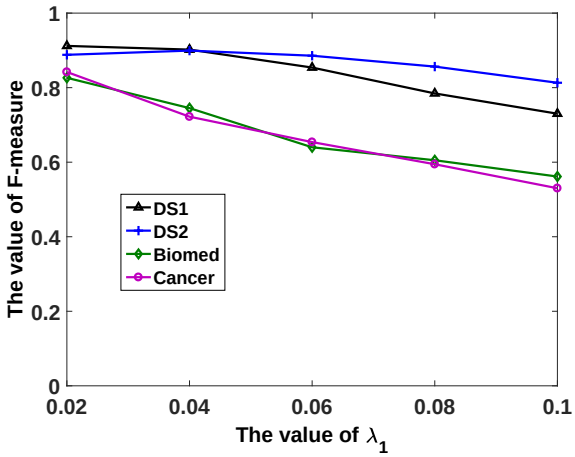


Fig. 11: The change of F-measure when setting different λ_1 values on some data sets.

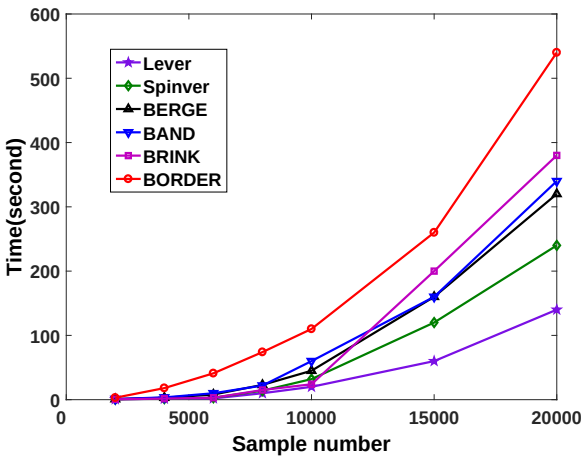


Fig. 13: The runtime of different algorithms with regard to the dataset size.

both synthetic datasets and real data sets demonstrate that our proposed model is effective and efficient.

Interdisciplinary thoughts may bring interesting ideas to the research on data mining. Analyzing the same problem from different perspectives may spark new solutions. How to detect the cluster boundary from more complex data, such as high-dimensional mixed-attribute data, from comprehensive perspectives will be our future work.

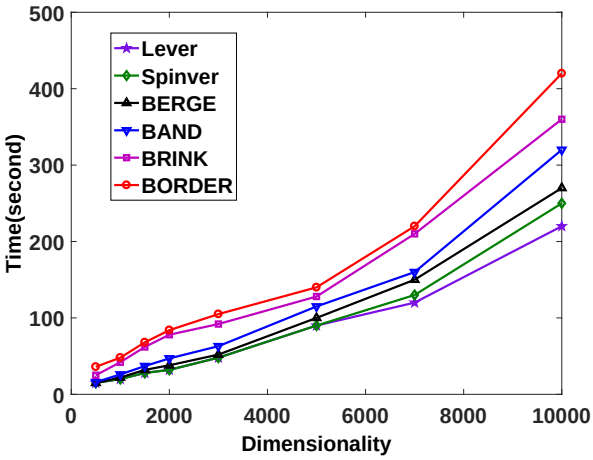


Fig. 14: The runtime of different algorithms with regard to the dimensionality of dataset.

REFERENCES

[1] E. Ergul, N. Arica, N. Ahuja, et al., "Clustering Through Hybrid Network Architecture With Support Vectors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no.6, pp. 1373 - 1385, 2017.

[2] X. Huang, Y. Ye, H. Zhang, "Extensions of Kmeans-Type Algorithms: A New Clustering Framework by Integrating Intracluster Compactness and Intercluster Separation," *IEEE Transactions on Neural Networks and Learning Systems*," vol. 25, no. 8, pp. 1433 - 1446, 2014.

[3] Z. Wang, Y.H Shao, L. Bai, N.Y Deng, "Twin Support Vector Machine for Clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2583 - 2588, 2015.

[4] A.K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recogn. Lett*, vol. 31, no. 8, pp. 651-666, 2010.

[5] Y.G. Lu and Y. Wan, "PHA: a fast potential-based hierarchical agglom-

erative clustering method," *Pattern Recognition*, vol. 46, no. 5, pp. 1227-1239, 2013.
[6] Y.W. Pang, S. Wang, Y. Yuan,"Learning Regularized LDA by Clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2191 - 2201, 2014.
[7] J. Goldberger, S. Gordon, and H. Greenspan," Unsupervised image-set clustering using an information theoretic framework," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 449-458, 2006.
[8] X.H. Hu, E. K. Park, and X.D. Zhang,"Microarray Gene Cluster Identification and Annotation Through Cluster Ensemble and EM-Based Informative Textual Summarization," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 5, pp. 832-840, 2009.
[9] Y.H. Zhao, J.X. Yu, G.R. Wang, L. Chen, B. Wang, and G. Yu,"Maximal Subspace Coregulated Gene Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 83-98, 2008.
[10] H. Bagher-Ebadian, H. Soltanian-Zadeh, S. Setayeshi, and S. T. Smith,"Neural network and fuzzy clustering approach for automatic diagnosis of coronary artery disease in nuclear medicine," *IEEE Transactions on Nuclear Science*, vol. 51, no. 1, pp. 184-192, 2004.
[11] F. Saadaoui, P. R. Bertrand, G. Boudet, and K. Rouffiac,"A Dimensionally Reduced Clustering Methodology for Heterogeneous Occupational Medicine Data Mining," *IEEE Transactions on NanoBioscience*, vol.14, no. 7, pp. 1707-715, 2015.
[12] Y.J. Horng, S.M. Chen, Y.C. Chang, and C.H. Lee,"A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 2, pp. 216-22, 2005.
[13] S. K. Bhatia and J. S. Deogun,"Conceptual clustering in information retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 3, pp. 427-436, 1998.
[14] N. Hohn, D. Veitch, and P. Abry,"Cluster processes: a natural language for network traffic," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2229-2244, 2003.
[15] K. Minkovich, C. Thibeault, M. O'Brien,"HRLSim: A High Performance Spiking Neural Network Simulator for GPGPU Clusters," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 316 - 331, 2014.
[16] J. Cao, L Li,"Cluster synchronization in an array of hybrid coupled neural networks with delay," *Neural Networks*, vol. 22, no. 4, pp. 335-342,20 09.
[17] H. Bassani, A. Araujo,"Dimension selective self-organizing maps with time-varying structure for subspace and projected clustering." *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 458 - 471, 2015.
[18] L. da Silva, D. Wunsch,"An Information-Theoretic-Cluster Visualization for Self-Organizing Maps," *IEEE Transactions on Neural Networks and Learning Systems* (2017), vol. PP, no. 99, pp. 1-19, 2017.
[19] J. Weston, F. Ratle, et al,"Deep learning via semi-supervised embedding." *Neural Networks: Tricks of the Trade*, pp. 639-655, 2012.
[20] C.Y. Xia, W. Hsu, M.L. Lee, et al.,"BORDER: An efficient computation of boundary points,"*IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 289-303, 2006.
[21] B.Z. Qiu, F. Yue, and J.Y. Shen,"BRIM: An efficient boundary points detecting algorithm," *Advances in Knowledge Discovery and Data Mining*, 2007.
[22] L.X. Xue and B. Z. Qiu,"Boundary Points Detection Algorithm Based on Coefficient of Variation," *Pattern Recognition and Artificial Intelligence*, vol. 22, no. 5, pp. 799-802, 2009.
[23] B.Z. Qiu, Y. Yang, and X.W. Du," BRINK: An Algorithm of Boundary Points of Clusters Detection Based On Local Qualitative Factors," *Journal of Zhengzhou University (Engineering Science)*,vol. 33, no. 3, pp. 117-121, 2012.
[24] L.X. Li, P. Geng, and B.Z. Qiu,"Clustering boundary detection technology for mixed attribute data set," *Kongzhi yu Juece/Control and Decision*, vol. 30, no. 1, pp. 171-175, 2015.
[25] B.Z. Qiu and X. Cao,"Clustering boundary detection for high dimensional space based on space inversion and Hopkins statistics," *Knowledge-Based Systems*, vol. 98, pp. 216225, 2016.
[26] M. Ester, H.P. Kriegel, J. Sander, et al.,"A density-based algorithm for discovering clusters in large spatial databases with noise,"*KDD*, pp. 226-231, 1996.
[27] J.J. Wu, H.F. Liu, H. Xiong, J. Cao, and J. Chen,"K-Means-Based Consensus Clustering: A Unified View,"*IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 155-169, 2015.
[28] S. Yu, L. Tranchevent, X.H. Liu, W. Glanzel, et al.,"Optimized Data Fusion for Kernel k-Means Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1031-1039, 2012.
[29] J. Yu and M.S. Yang,"Optimality test for generalized FCM and its application to parameter selection,"*IEEE Transactions on Fuzzy Systems*, vol. 13, no. 1, pp. 164-176, 2005.
[30] J. Yu, Q.S. Cheng, and H.K. Huang,"Analysis of the weighting exponent in the FCM," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 634-639, 2004.
[31] P. Emanuel,"On the estimation of a probability density function and the mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065-1076, 1962.
[32] K. Fukunaga and L.D. Hostetler,"The Estimation of the Gradient of a Density Function," *IEEE Trans. Information Theory*, vol. 21, no. 1, pp. 32-40, 1975.
[33] Y.Z. Cheng,"MeanShift, Mode seeking, and Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790-799, 1995.
[34] J. Li, H.F. Chen, G. Li, B. He, et al.,"Salient object detection based on meanshift filtering and fusion of colour information," *IET Image Processing*, vol. 9, no. 11, pp. 977-985, 2015.
[35] M. Kleider, B. Rafaely, B. Weiss, and E. Bachmat,"Golden-Ratio Sampling for Scanning Circular Microphone Arrays," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2091-2098, 2010.
[36] A. Hinneburg and D.A Keim,"Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering," *Very Large Data Bases (VLDB)*, pp. 506-517,1999.
[37] J. Hou, H.J. Gao; Q. Xia, et al.," Feature Combination and the kNN Framework in Object Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1368 - 1378, 2016.
[38] S.C. Zhang, X.L Li, M. Zong, et al.,"Efficient kNN Classification With Different Numbers of Nearest Neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 9, pp. 1 - 12, 2017.
[39] H. Samet,"K-Nearest Neighbor Finding Using MaxNearestDist," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 243-252, 2008.
[40] S.L Liu, L. F, H. Qiao,"Scatter Balance: An Angle-Based Supervised Dimensionality Reduction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 2, pp. 277 - 289, 2015.
[41] Z.W. Yu, P.N. Luo, et al.,"Incremental Semi-Supervised Clustering Ensemble for High Dimensional Data Clustering,"*IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 701-714, 2016.
[42] E. Cesario, G. Manco, and R. Ortale,"Top-Down Parameter-Free Clustering of High-Dimensional Categorical Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1607-1624, 2007.
[43] L.P. Jing, M. K. Ng, and J.Z. Huang,"An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data,"*IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, pp. 1026-1041, 2007.
[44] F. Schmidt, A. Vikhlinin, and W. Hu,"Cluster Constraints on f(R) Gravity," *Physical Review D Particles & Fields*, vol. 80, no. 8, pp. 350-356, 2009.
[45] S. Kundu,"Gravitational clustering: a new approach based on the spatial distribution of the points," *Pattern recognition*, vol. 32, no. 7, pp. 1149-1160, 1999.
[46] P. Lingras, M. Chen, and D.Q. Miao,"Rough Cluster Quality Index Based on Decision Theory," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21 no. 7, pp. 1014 -1026, 2009.
[47] O. Loyola, M. A. Medina, and M. Garcia,"Inducing Decision Trees based on a Cluster Quality Index," *IEEE Latin America Transactions*, vol. 13 no. 4, pp. 1141-1147, 2015.
[48] G. Hripcsak and A.S. Rothschild,"Agreement, the F-Measure, and Reliability in Information Retrieval," *Journal of the American Medical Informatics Association Jamia*, vol. 12 no. 3, pp. 296-298, 2005.
[49] L. Liu and M. T. zsu,"Encyclopedia of Database Systems,"Springer, 2009.
[50] http://lib.stat.cmu.edu/datasets/biomed.data.html.
[51] http://archive.ics.uci.edu/ml/datasets.html
[52] http://genomics-pubs.princeton.edu/oncology/affydata/
[53] D.X. Jiang, C. Tang, and A.D. Zhang," Cluster analysis for gene expression data: A survey,"*IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370-1386, 2004.
[54] http://www.gems-system.org/
[55] J.S. Wang and F.C. Chuang,"An Accelerometer-Based Digital Pen With a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 2998-3007, 2012.

[56] Z.K Lu, Z.R. Chi, and W.C. Siu. " Extraction and optimization of B-spline PBD templates for recognition of connected handwritten digit strings,"*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 132-139, 2002.

[57] http://yann.lecun.com/exdb/mnist.html

[58] J. Soldera, C.A.R. Behaine, and J. Scharcanski. "Customized Orthogonal Locality Preserving Projections With Soft-Margin Maximization for Face Recognition", *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 9, pp. 2417-2426, 2015.

[59] B.F. Klare, M.J. Burge, et al.,"Face Recognition Performance: Role of Demographic Information," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1789-1801, 2012.

[60] P. P. Brahma, D.P. Wu, Y.Y. She,"Why Deep Learning Works: A Manifold Disentanglement Perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 10, pp. 1997 - 2008, 2016.

[61] K.K. Huang, D.Q. Dai, et al.,"Learning Kernel Extended Dictionary for Face Recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1082 - 1094, 2017.

[62] http://www-prima.inrialpes.fr/Pointing04

[63] X. Liu, D.C. Tao, M. L, et al.,"Learning to Track Multiple Targets," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1060 - 1073, 2015.

[64] X.Z. Wang and D. Musicki,"Low elevation sea-surface target tracking using IPDA type filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 759-774, 2007.

[65] R. C. Luo and T. M. Chen,"Autonomous mobile target tracking system based on grey-fuzzy control algorithm," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 4, pp. 920-931, 2000.

[66] http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm

[67] M.M. Breunig,"LOF: identifying density-based local outliers," *ACM SIGMOD Record*, 2000.

[68] W. Jin,"Ranking outliers using symmetric neighborhood relationship," *Advances in Knowledge Discovery and Data Mining*, Springer, pp. 577593, 2006.

[69] B. Du, C. Liu, W. Zhou, Z. Hou, and H. Xiong,"Catch Me If You Can: Detecting Pickpocket Suspects from Large-Scale Transit Records," *KDD*, 2016.

[70] J. Liu, L. Sun, W. Chen, and H. Xiong,"Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization," *KDD*, 2016.

[71] J. Huang, Q. Zhu, L. Yang, and J. Feng,"A non-parameter outlier detection algorithm based on natural neighbor," *Knowledge-Based Systems*, vol. 92, pp. 71-77, 2016.

[72] T.N. Raymond and J. Han,"Efficient and effictive clustering methods for spatial data mining," *VLDB*, 1994.

[73] T. Zhang, R. Ramakrishnan, and M. Livny,"BIRCH: An efficient data clustering method for very large databases,"SIGMOD, 1996.

[74] S. Ruggieri,"Efficient C4.5," *IEEE Transactions on Knowledge a Data Engineer*, vol. 14, no. 2, pp. 438-444, 2002.

[75] Q. Wu and R. Law,"Fuzzy support vector regression machine with penalizing Gaussian noises on triangular fuzzy number space," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7788-7795, 2010.

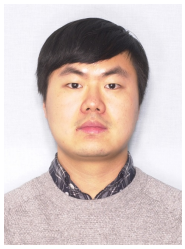
[76] R. Maini and H. Aggarwal,"Study and Comparison of Various Image Edge Detection Techniques," *International Journal of Image Processing*, vol. 3, no. 1, pp. 1-11, 2009.

[77] Q.M. Peng, Y.M. Cheung, X.G. You, and Y.Y. Tang, "A Hybrid of Local and Global Saliencies for Detecting Image Salient Region and Appearance", *IEEE Transactions on Systems, Man and Cybernetics: Systems*, DOI: 10.1109/TSMC.2016.2564922, 2016.

Xiaofeng Cao received his B.E. degree and M.S. degree in Zhengzhou University, Zhengzhou, China, in 2014 and 2017, respectively. He is currently a PhD candidate at Advanced Analytics Institute, University of Technology Sydney. His research interests include machine learning and data mining.

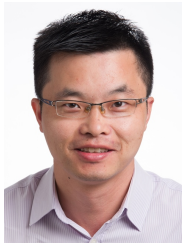
Baozhi Qiu is a Professor and Director of Data Mining & Machine Learning Laboratory at School of Information Engineering, Zhengzhou University. He received Ph.D. in Computer Science from Xi'an Jiaotong University in 2006. His research interests include data mining, machine learning, and database application.

Lixiang Li is a professor in the School of Information Engineering, Zhengzhou University, Zhengzhou, China. She received her Master's degree in computer science from Xian Jiaotong University in 1996. Her research interests include data mining and computer network.



Zenglin Shi received his B.E. degree and M.S. degree in Zhengzhou University, Zhengzhou, China, in 2014 and 2017, respectively. He is currently a PhD candidate at the faculty of Science, University of Amsterdam. His research interest covers computer vision, machine learning, and deep learning.

Guandong Xu is Associate Professor and Program Leader at School of Software and Advanced Analytics Institute, University of Technology Sydney and he received PhD degree in Computer Science from Victoria University, Australia. His research interests cover Data Science, Data Analytics, Recommender Systems, Web Mining, User Modelling, NLP, Social Network Analysis, and Social Media Mining. He has published three monographs in Springer and CRC press, and 180+ journal and conference papers including TOIS, TIST, TNNLS, TSC, TIFS, IEEE-IS, Inf. Sci., KAIS, WWWJ, KBS, Neurocomputing, ESWA, Inf. Retr., IJCAI, AAAI, WWW, ICDM, ICDE, CIKM. He is the assistant Editor-in-Chief of World Wide Web Journal and has been serving in editorial board or as guest editors for several international journals, such as Social Network Analysis and Mining, the Computer Journal, Journal of Systems and Software, World Wide Web Journal, Multimedia Tools and Applications, and Online Information Review.



Jianliang Xu is a Professor in the Department of Computer Science, Hong Kong Baptist University. His research interests include big data analytics, mobile computing, and data security and privacy. He has published more than 150 technical papers in these areas. He has served as a program cochair/vice chair for a number of major international conferences including IEEE ICDCS 2012, IEEE CPSNA 2015, APWeb-WAIM 2018, and IEEE MDM 2019. He is an associate editor of the IEEE Transactions on Knowledge and Data Engineering (2014-) and the Proceedings of the VLDB Endowment (2018).