Neurocomputing 000 (2017) 1–9

Contents lists available at ScienceDirect

Neurocomputing

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

A clustering algorithm using skewness-based boundary detection

Xiangli Li, Qiong Han*, Baozhi Qiu

School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China

ARTICLE INFO

Article history:
Received 7 April 2016
Revised 17 December 2016
Accepted 4 September 2017
Available online xxx

Communicated by zhi yong Liu

Keywords:
Skewness
Boundary degree
Clustering algorithm
Clustering boundary

ABSTRACT

Clustering analysis has been applied in all aspects of data mining. Density-based and grid-based clustering algorithms are used to form clusters from the core points or dense grids to extend to the boundary of the clusters. However, deficiencies are still existed. To find out the right boundary and improve the precision of the cluster, this paper has proposed a new clustering algorithm (named C-USB) based on the skew characteristic of the data distribution in the cluster margin region. The boundary degree calculated by skew degree and the local density are used to distinguish whether a data is an internal point or non-internal point. And the connected matrix is constructed by removing the neighbor relationships of non-internal points from the relationships of all points, then the clusters can be formed by searching from the connected matrix towards internal of the clusters. Experimental results on synthetic and real data sets show that the C-USB has higher accuracy than that of similar algorithms.

1. Introduction

Clustering refers to a process to discover the internal structures of data or the potential data models in a dataset [1–3] by data partitioning. Thanks to the outstanding capability of discover clusters of different shapes and sizes along with outliers, density-based [4–6] and grid-based [7] clustering technology are widely applied to the fields of health care [8], information security [9], internet [10] and etc [11–15].

Data points are divided into core points, boundary points and noise points by the DBSCAN algorithm [16], and a cluster is formed when the data is expanding from the core points outwards the clustering boundary. As those methods are susceptible to parameter changes, different parameters may lead to different data dividing and clustering results. IS-DBSCAN [17], ISB-DBSCAN [18] and others [19–21] are proposed by making use of the nearest neighbor relationship instead of the neighborhood density, which effectively reduce the influence of the parameters on the algorithm. However, for the multi-density datasets, the clustering results are not always favorable because neighbor relationships can misjudge the boundary points.

Grid clustering technique divide grids into high-density ones and low-density ones with compressing expression and clusters are formed when high-density grids are connected. Grid-based clustering technologies, such as CLIQUE [22], MGM-GA [23] and etc [24,25], are efficient because grid clustering is formed with the extension of grid cells. Such an approach can be efficient, however, in the clusters forming process, if a dense grid is adjacent to a sparse grid (we called boundary grid), which probably contains noises, the algorithm is of low clustering accuracy.

Boundary points not only play a significant role in expansion-based clustering algorithms, but also in other fields of data mining. The PAC-Bayes boundary theory, a theoretical framework, combines Bayes theory [26–28] with minimum structural analysis principle of random classifier, obtaining the most generalized risk boundary. The algorithms derived from PAC-Bayes boundary are actually the “average” of Hypothesis Space, thus achieving a better classification performance [29–31]. Support Vector Machine (SVM) [32–34] also uses boundary points to improve performance. Furthermore, on the occasion of supervised, Compression Nearest Neighbor (CNN) [35] can extract the neighboring data boundary points from different classes, and it can also used to reduce the number of support vectors in the SVM algorithm, which is helpful to reduce training costs [36–39]. Besides, the study on boundary is also contributing to discover interesting models in data [40–42]. For instance, in the medical field, the clustering boundary may represent a group of people, who carry virus but not affected. With regard to handwriting recognition, the clustering boundary may stand for handwriting images which are easily misjudged to be other characters.

2. Motivation

From the standpoint of density-based clustering technology, clusters refer to the dense regions separated by sparse regions. For

* Corresponding author.
E-mail address: qhanzzu@163.com (Q. Han).

https://doi.org/10.1016/j.neucom.2017.09.023
0925-2312/

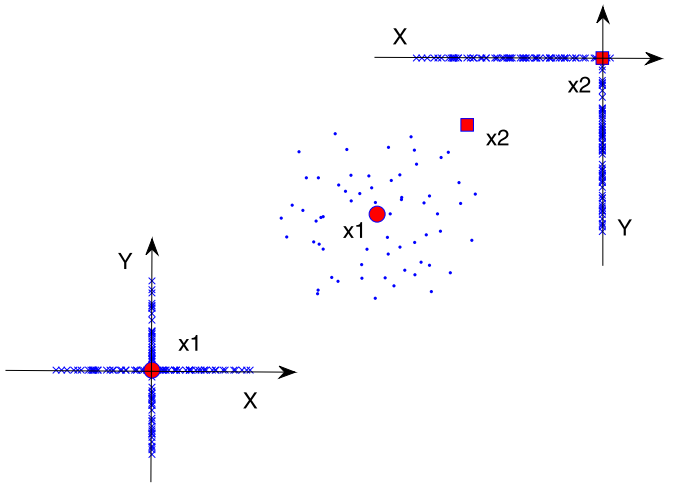

Fig. 1. Point x_1, x_2 and distribution of points after mapping on coordinate axis.

grid-based clustering technology, clusters are formed by the connected high-density grid units. Both of the technologies are adopting the expansion-based method to form clusters from core points (dense grids) to boundary points (boundary grids). And the similar clustering pattern makes it possible for the combination of them to solve cluster problems [43–45]. Boundary points, as the terminal condition of expansion, are of great importance to both technologies. In density-based clustering algorithm, the boundary points are identified by density only. However, in real datasets, margin density is probably equal to or larger than that of the internal region. Grid-based clustering algorithm determines boundaries by statistics of points within a single grid, which always causes the internal points or noise points which located nearby the boundary grid to be misjudged as boundary points.

Based on huge data analysis, internal points are founded to be surrounded by their neighboring points, while the neighboring points of boundary point are always located in one side of it. Fig. 1 shows that, x_1 and x_2 are taken as reference points respectively, and the points around them are mapped into X and Y axis. After the mapping of central x_1, other points can be found to symmetrically spread on the two sides of x_1 in different axis after mapping. However, when x_2 is regarded as the reference point, other points are located in just one side of x_2 in X or Y axis only after mapping; thus, the mappings of point x_2 lie in a skew pattern. Based on the above features, this paper proposes a new skewness-based measurement method to separate boundary objects from a dataset. Unlike the density-based or grid-based method, boundary points in this method are determined by the distribution of their neighbors, which can effectively avoid the effects of density and neighboring radius.

3. Algorithm

3.1. Skewness and boundary degree

Suppose a dataset $X = \{x_i | x_i \in R^{m*n}; i = 1, 2, ..., n; m, n \in N\}$.

Definition 1 ($Skewness\, S_c(x_p)$)**.** $Skewness\, S_c(x_p)$ is used to measure the data skew distribution, defined as follows:

$$S_c(x_p) = \frac{\sum_{j=1}^{m}\sum_{i=1}^{k}(x_{ij} - x_{pj})^2}{k} \qquad x_{ij} \in N_{k-dist}(x_p). \tag{1}$$

Among which $N_{k-dist}(x_p)$ [46] refers to k nearest neighbor of x_p, $x_p \in X$.

Skewness has been widely utilized in the field of data statistics and analysis [47–49]. This paper aims to study whether the distribution of neighboring points of x_p are skew or not. Therefore, x_p itself is taken as the reference point. Left skewness or right skewness has no impact on the study, so the definition of $S_c(x_p)$ is changed to a non-directional value.

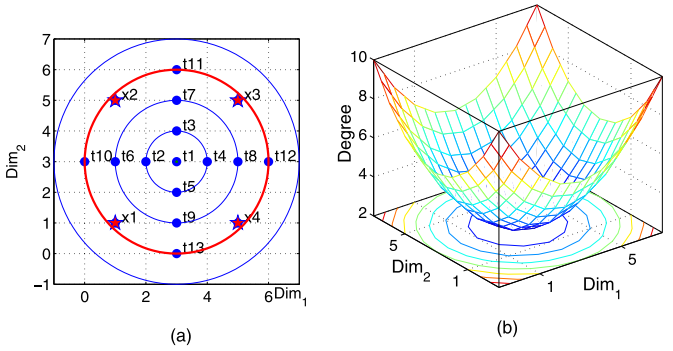

Fig. 2. Boundary degree testing model. (a) Points $x_1 - x_4$ and Positions $t_1 - t_{13}$ (b) 3-dim view of boundary degree.

Definition 2 (Local density $D_{local}(x_p)$)**.** The *local density* refers to the compactedness of point x_p and its neighboring points. *Local density* is the reciprocal of summing up standard deviations drawn from different dimensions of x_p and its neighboring points. The definition is as follow:

$$D_{local}(x_p) = \frac{1}{\sum_{j=1}^{m}\sqrt{\frac{1}{k}\sum_{i=1}^{k}(x_{ij} - \bar{x})^2}} \qquad x_{ij} \in N_{k-dist}(x_p). \tag{2}$$

$\bar{x}$ refers to the average value of x_p and its k nearest neighbors. In the sparse areas, the value of $D_{local}(x_p)$ is relatively small. In the dense areas, the value of $D_{local}(x_p)$ is relatively large.

Definition 3 (Boundary Degree)**.** *Boundary degree* refers to the degree of boundary of data points. The *boundary degree* of x_p is calculated by *skewness* $Sc(x_p)$ multiplying *local density* $D_{local}(x_p)$. The definition is as follow:

$$Boundary\, degree = S_c(x_p)D_{local}(x_p) = \frac{\sum_{j=1}^{m}\sum_{i=1}^{k}(x_{ij} - x_{pj})^2}{k\sum_{j=1}^{m}\sqrt{\frac{1}{k}\sum_{i=1}^{k}(x_{ij} - \bar{x})^2}}. \tag{3}$$

Where, $\bar{x}$ refers to the average value of x_p and its k nearest neighbors. The smaller the *boundary degree* of data point x_p is, the closer x_p gets to the central position of a cluster. But when the *boundary degree* gets larger, x_p will get closer to or be on the cluster margin position of a cluster.

3.2. Discussions on boundary degree

To clearly and quantitatively study the changes of boundary degree, this paper has examined the features of boundary degree with modeling. Suppose that x_1, x_2, x_3, x_4 are the neighboring points of point p and point p is movable. Fig. 2(a) shows that, t_1 is located in the central position of x_1, x_2, x_3, x_4. Expand t_1 to get three groups of positions, including (t_2, t_3, t_4, t_5), (t_6, t_7, t_8, t_9) and (t_{10}, t_{11}, t_{12}, t_{13}). The circumradius of the three groups is apart with a distance unit from one to another. (t_{10}, t_{11}, t_{12}, t_{13}) are located on the margin position of a circumcircle formed by x_1, x_2, x_3, x_4. In accordance with the skewness definition, the value of boundary degree should increase from t_1 to t_{13}. Upon calculation, the boundary degree value at relevant positions is shown in Table 1, while the 3-dimensional graph of boundary degree is shown in Fig. 2(b).

From Table 1, the value of boundary degree of t_1 is the smallest. Based on the position of t_1, $t_2 - t_5$ move outwards with a

Please cite this article as: X. Li et al., A clustering algorithm using skewness-based boundary detection, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.023

Table 1
Boundary degree of $t_1 - t_{13}$.

i	t_i	$S_c(t_{i1})$	$S_c(t_{i2})$	$S_c(t_i)$	$D_{local}(t_i)$	$Degree(t_i)$
1	(3,3)	16	16	8	0.25	2.00
2	(2,3)	20	16	9	0.25	2.25
3	(3,4)	16	20	9	0.25	2.25
4	(4,3)	20	16	9	0.25	2.25
5	(3,2)	16	20	9	0.25	2.25
6	(1,3)	32	16	12	0.25	3.00
7	(3,5)	16	32	12	0.25	3.00
8	(5,3)	32	16	12	0.25	3.00
9	(3,1)	16	32	12	0.25	3.00
10	(0,3)	52	16	17	0.25	4.25
11	(3,6)	16	52	17	0.25	4.25
12	(6,3)	52	16	17	0.25	4.25
13	(3,0)	16	52	17	0.25	4.25

distance unit, the boundary degree values is 2.25, with an increase of 0.25, rising by 12.5%; $t_6 - t_9$ move outwards with a distance unit from $t_2 - t_5$, the boundary degree values is 3, with an increase of 0.75, rising by 33.33%; whereas $t_{10} - t_{13}$ move outwards from $t_6 - t_9$ with a distance unit, boundary degree values of them are 4.25, with an increase of 1.25, rise 41.67%. In regard to the same neighboring points, boundary degrees increase quickly along with $t_2 - t_5$, $t_6 - t_9$, $t_{10} - t_{13}$, which is consistent with the conclusion in the Fig. 2(b) that the boundary degrees increase and change rapidly from the central position towards the outside.

3.3. Connection matrix

Definition 4 (threshold α)**.** *Threshold* α divides points into internal points and non-internal points, functioning as a percentile after the order descending of boundary degree.

Definition 5 (Connection matrix $Conn(i, j)$)**.** $Conn(i, j)$ shows the relationship of point x_i and point x_j. Which is defined as follows:

$$Conn(i, j) = \begin{cases} 1 & x_i \in N_{k-dist}(x_j) \\ 0 & x_i \notin N_{k-dist}(x_j) \end{cases}. \quad (4)$$

If point x_i is the k nearest neighbor point of x_j, the value of $Conn(i, j)$ will be 1 ; otherwise, $Conn(i, j)$ will be 0. So $Conn$ is a $n \times n$ matrix.

Definition 6 (Connection matrix $SubConn(i, j)$)**.** $SubConn(i, j)$ shows the relationship of point x_i and point x_j. $SubConn(i, j)$ will be obtained when removing the neighbor relationships of non-internal points from $Conn(i, j)$. That is:

$$SubConn(i, j) = \begin{cases} Conn(i, j) & x_i \text{ is internal}, \ x_i \in N_{k-dist}(x_j) \\ 0 & x_i \text{ is not internal}, \ x_i \in X \end{cases}. \quad (5)$$

In the paper, Fig. 3 directly represents the differences between the two matrixes. Fig. 3(a) shows Conn and each point in Fig. 3(a) has 5 nearest neighbors. As indicated in the picture, there are three target clusters, which are actually connected by shared sides linked by certain points. SubConn is shown in Fig. 3(b). After removing neighborhood relations of non-internal points (red points in the picture), the three clusters are separated from each another, which is consistent with the actual conditions. Moreover, boundary points are situated on cluster margins, while noise points still exist individually.

Dataset Syn1 [50] (Fig. 4(a)), contains 6 target clusters and a large amount of noise points and interfering lines. Boundary degrees after ordering are displayed in Fig. 4(b), while the corresponding data points of three intervals are shown in Fig. 4(c)(d)(e).

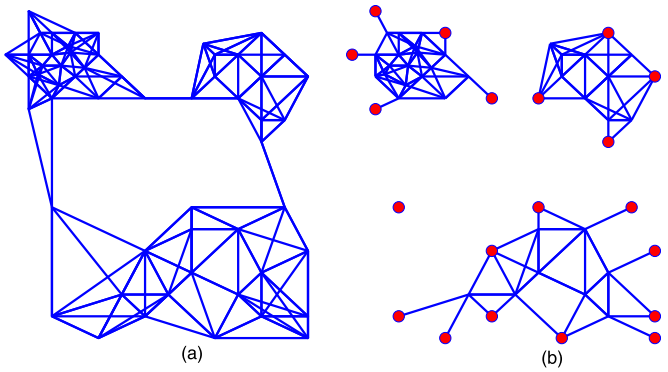

Fig. 3. Differences between Conn and SubConn. (a) Conn (b) SubConn.

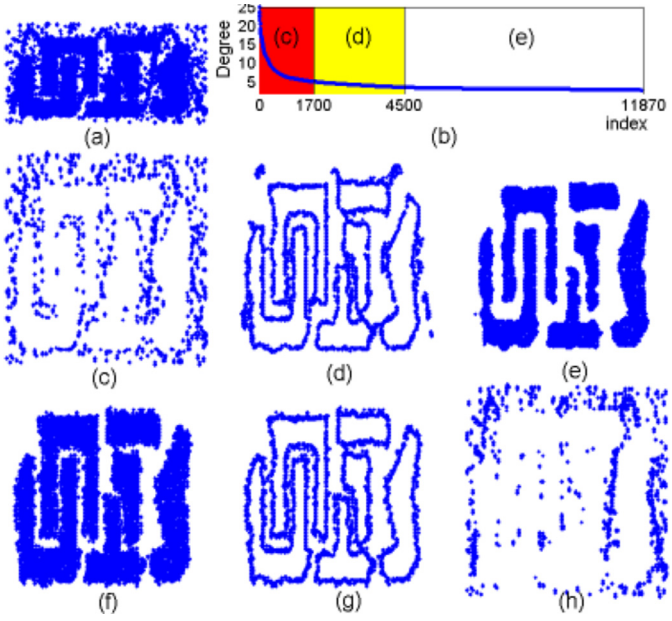

Fig. 4. Examples of Syn1. (a) data set (b) boundary degree ordering (c), (d), (e) corresponding points of the three ranges (f) clustering results (g) boundary points (h) noise points.

As shown in the three pictures, boundary degrees are able to distinguish noise points and interfering lines from data set. We can tell the outline of the clusters in Fig. 4(d), but there are still some noises in it. Therefore, it is not proper to identify cluster boundaries by this mean.

The difference between boundary points and noise points is that: boundary points are within the range of clusters, while noise points drift away from clusters. Supposed $\alpha = 37.94\%$, we build SubConn and conduct graph search, and display the subgraph in which the number of its object is bigger than $k = 20$ in Fig. 4(f). Non-internal points within Fig. 4(f) are presented in Fig.4(g). And the points not in Fig. 4(f) are shown in Fig. 4(h). In comparison with Fig. 4(d), Fig. 4(g) removes misjudged noise points from boundary points.

3.4. Algorithm description

The algorithm first calculates the boundary degree of all points, and structure SubConn with threshold α, and at last perform subgraph searches to get clustering results.

Algorithm: C-USB

Input: dataset, the number of neighbors k, threshold α

Please cite this article as: X. Li et al., A clustering algorithm using skewness-based boundary detection, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.023

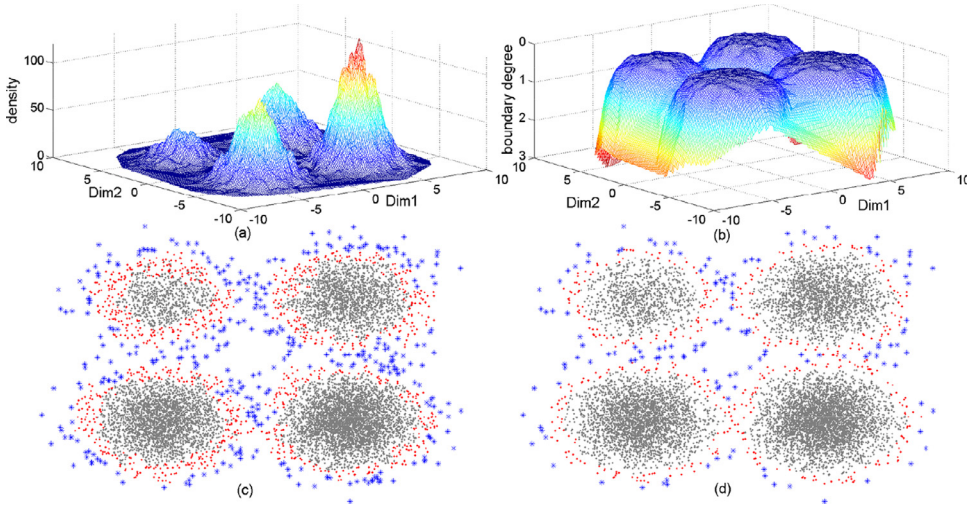

Fig. 5. The comparison of DBSCAN and C-USB on XOR. (a) Density (b) Boundary degree (c) Result of DBSCAN (d) Result of C-USB.

Output: clustering results, common boundary, singular boundary, noises

Steps:

1. Calculate k nearest neighbors of point p and its boundary degree, as shown in Definition 3.
2. Divide data into internal points and non-internal points with threshold α, as shown in Definition 4.
3. Structure SubConn and conduct sub-graph searches, to obtain clustering results, as shown in Definition 6. The intersections of non-internal point gathering and point sets in all clusters are common boundary, whose intersection with a cluster is singular boundary. The points in the non-clusters are noise points.

4. Experiments and analysis

4.1. Experimental design and environment

As the mainline of the experiment, datasets are selected from low to high dimensions to test the performance of the algorithm on different dimensions. The experiment is conducted on a MATLAB 2012 computer equipped with Intel Core 2.93GHZ CPU, 4G RAM, and Window 7 operation system.

4.2. XOR dataset

As the two-dimensional dataset generated by MATLAB, XOR consists of four clusters of normal distribution with 10,000 points (1000, 2000, 3000, 4000, respectively). The target cluster sizes and densities of the target clusters are different from each other, and the points scattering in the edge area are intertwined, as shown Fig. 5(a). This paper compares C-USB with DBSCAN to elaborate on their differences. The experimental results are shown in Fig. 5.

Refer to Fig. 5(a) and (b) for corresponding density and boundary degree of data points. Fig. 5(a) is in a cone shape, that means, density on cluster edges reports the smaller value with mild changes, while registering the bigger value at the core with rigorous changes. This will lead to select a great number of data points on the edge with the same density value, which is negative to select threshold to separate boundary and internal points. The values of boundary degree display to be an upside-down drop. In the internal areas, there is little impact of multi-density on boundary values, which change slightly. In the fringe area, significant changes are detected in boundary degree, favorable for choosing threshold. The cluster results drawn from the two algorithms are shown as Fig. 5(c) and (d), where gray points are internal, red is boundary point, and blue is noise point. The internal and boundary points included in clusters are marked with different colors for the sake of visual comparison. As shown in Fig. 5, the boundary points of C-USB are reported to be 456, with a reduction of 56.49%, to 592, less than DBSCAN algorithm. In this paper, the C-USB algorithm utilizes fewer points to outline the boundary, as ways to reflect the actual margin of clusters. Density-based clustering technology considers that clustering is able to separate the high-density areas in the low-density regions. Meanwhile, Support Vector Machine (SVM) believes that clusters can be divided by hyperplane. Low density areas and hyperplane correspond to boundary points of clusters, for instance, the support vector could converged by boundary points. On the occasion of unsupervised, the algorithm C-USB is able to detect boundary points of clustering and effectively reduce their number. When establishing a hyperplane with boundary points rather than all points, the algorithm can remarkably decrease the number of support vector to reduce training hyperplane costs.

4.3. Dataset Syn2

Dataset Syn2 consists of 5 clusters in different densities, a little amount of noises, and a short bridge, which cases clustering algorithm not to discover the correct cluster numbers.

IS-DBSCAN is a typical algorithm to identify density by neighbor relationship, which determines non-internal points if IS_k is smaller than the value of $2k/3$. This reduces its parameters, but cannot always obtain the best solutions, particularly, when processing datasets with noises and interference, it cannot effectively adjust the threshold for internal and non-internal points, causing it not to correctly identify the number of clusters. Fig. 6 shows the experimental results of IS-DBSCAN and C-USB.

Refer to IS_k and boundary degree of points in Fig. 6(a) and (b). The value of IS_k is finite integer. The narrow value range make some boundary and internal points select same values, which is not good for the selection of threshold. Boundary degree on the non-edge of clusters are out of reach of the impact of cluster sizes and density, and changes greatly on the edge areas. This shows that the boundary degree is sensitive to boundary points, while insensitive to internal points. From Fig. 6(c) and (d), boundary points of IS-DBSCAN cannot correctly report the outlines of cluster margin, causing many cluster internal points to be misjudged to be boundary points. IS-DBSCAN cannot correctly identify short bridges, giving rise to only 4 clusters being found (two clusters on

Please cite this article as: X. Li et al., A clustering algorithm using skewness-based boundary detection, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.023

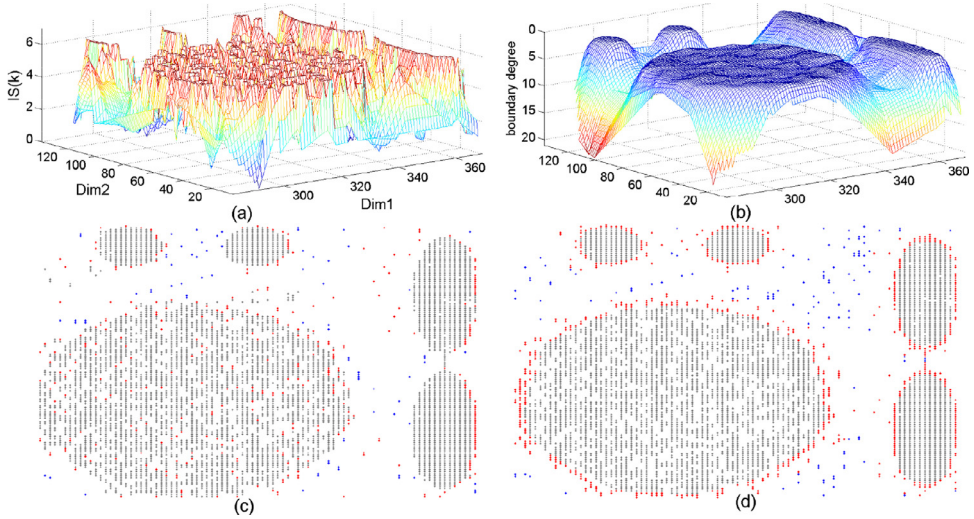

Fig. 6. Comparison of IS-DBSCAN and C-USB on Syn2. (a) IS_k (b) Boundary degree (c) Results of IS-DBSCAN (d) Results of C-USB.

Table 2
Information of image datasets.

Dataset	*Form*	*No.*	*Clusters*
Data1	28×28	1135,1028	Num. 1,7
Data2	28×28	892,6265	Num. 5,7
Data3	30×30	16,16,16,16,33	Acer:Palmatum,Pictum,tanoids,ubrum,noise
Data4	90×80	33,34,72,58,23,16	S084,S086,S088,S135,S134,S089

the right are misclassified to be the same one). However, boundary points in the algorithm of the article are able to accurately describe the cluster outlines and effectively prevent short bridges, to discover 5 clusters in total.

4.4. Image dataset

The analysis on high-dimensional data has been the bottleneck in data mining [51,52]. Although its impalpable to understand high-dimensional data, human is able to distinguish the differences among images. In this paper, each pixel point is deemed to be one-dimensional data, while translated images as high-dimensional data [53,54]. Some image datasets are organized to test the clustering performance of the algorithm in high-dimensional data.

The image datasets in the paper are from UCI standardized dataset. MNIST dataset, Leaves plant species dataset, and Cohn_kanade dataset include many sub-datasets, which are selected and translated into fixed format to build datasets, which is shown in Table 2.

The algorithm results on data1 are shown in Fig. 7, which contains of digital 1 cluster, digital 7 cluster, and a small amount of noises. The cluster boundary presents on the outermost of clusters.

For the purposes of convenient uses and easy understanding, the writing of figures keeps to be simple, convenient, neatly, and cognizable. The dataset combined with randomly-selected figures should be in normal distribution. That means, neat writing should account for the majority of the dataset, the minority of untidy but cognizable writing, and even less indistinguishable. The results drawn out from the algorithm is consistent with the distribution principle. Internal points (standardized writing samples) contribute to 92.46% of the dataset, boundary points (untidy but cognizable samples) are of 4.3%, and noise points (indistinguishable writing samples) only take 3.24%. It can be seen that, when writing in the daily life, figures written in the pattern of boundary points in the picture, will be recognized to be other digital numbers.

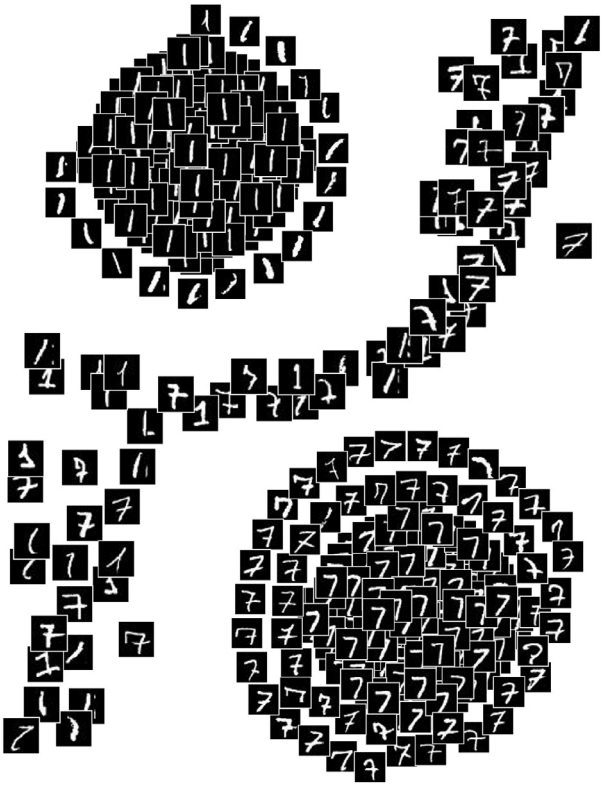

Fig. 7. The Results of C-USB on Data1.

Data3 dataset consists of four target clusters (16 points of each) and noise points (33). Different algorithms will process boundary and noise in different ways. See Fig. 8 for the experimental results of K-means++, IS-DBSCAN, ISB-DBSCAN, and C-USB, respectively (noise points are shown in corresponding corners).

Please cite this article as: X. Li et al., A clustering algorithm using skewness-based boundary detection, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.023

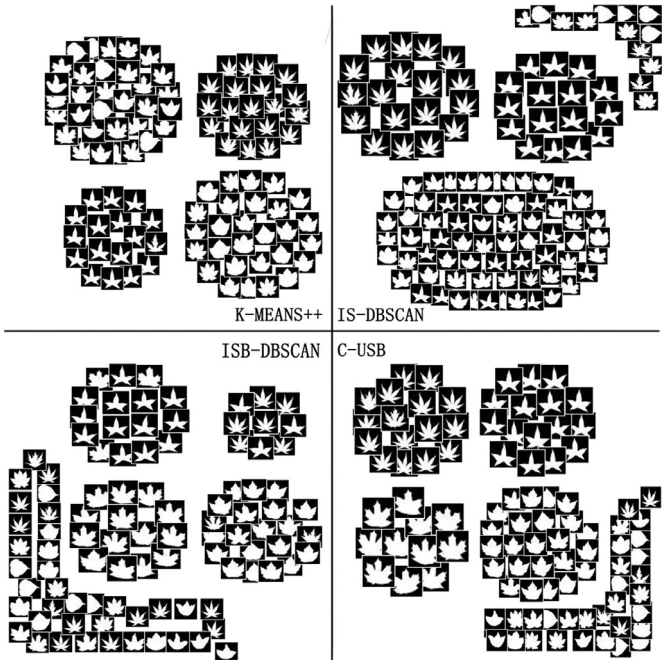

Fig. 8. Results comparison of different algorithms on data4.

Table 3
Algorithm results comparison on Data1-4.

Dataset	Algorithm	Accuracy (%)	F-measure (%)	Entropy	Purity
Data1	K-means++	95.61	97.76	0.2333	0.9561
	IS-DBSCAN	99.90	96.31	0.0103	0.9990
	ISB-DBSCAN	99.95	92.93	0.0062	0.9995
	C-USB	99.95	92.64	0.0059	0.9995
Data2	K-means++	55.06	71.02	0.5242	0.8752
	IS-DBSCAN	99.86	95.22	0.0121	0.9986
	ISB-DBSCAN	99.76	89.05	0.0201	0.9976
	C-USB	99.92	94.53	0.0053	0.9995
Data3	K-means++	56.70	72.37	1.0866	0.6495
	IS-DBSCAN	57.14	68.85	1.0469	0.6071
	ISB-DBSCAN	73.33	67.11	0.9626	0.7113
	C-USB	85.07	76.24	0.5727	0.8507
Data4	K-means++	89.62	94.53	0.2362	0.9322
	IS-DBSCAN	100.0	92.24	0	1
	ISB-DBSCAN	80.38	81.27	0.5366	0.8038
	C-USB	100.0	96.49	0	1

As shown in Fig. 8, different algorithms enjoy unique features. K-means++ algorithm is of low accuracy for incapacity of de-noising. Although IS-DBSCAN and ISB-DBSCAN have the function of de-noising, they differ from each other. The former algorithm identifies only less points to be noises and only find 3 clusters, while the latter determines more points as noises and discovers four clusters. The algorithm in the paper is able to accurately find clusters, and judge the number of noises as accurate as possible.

The algorithm assessment on Data1-4 refers to Table 3.

From Table 3, C-USB enjoys great advantages in clustering accuracy, entropy evaluation, and purity. In the F-measure evaluation in Data1 and Data2, the algorithm is not considered to be the best way, as it is connected with the features of assessment criteria. The accuracy and recall rate of clustering algorithm are the two sides of a coin. Better accuracy gives rise to low recall rate, vice versa. F-measure integrates the two indicators, showing that the value of F-measure will be larger, only if both accuracy and recall rate are set with higher values. The C-USB is able to accurately distinguish internal and non-internal points, which relevantly reduces the recall rate, causing F-measure value probably not to be optimal. However, it will improve the purity and accuracy of cluster results, which symbolizes the priority for clustering.

Table 4
Information of UCI datasets.

Dataset	*No.*	*Dim*	*No.ofClusters*
Breast cancer	699	9	458,241
Seeds	210	7	70,70,70
Heart statling	270	13	120,150
Diabetes	768	8	268,500
Chart time	600	60	100,100,100,100,100,100

4.5. UCI DataSet

The UCI database, founded by University of California Irvine, is used for machine learning and has been frequently-used and standardized testing dataset in the field. Five high-dimensional datasets —Breast cancer, Seeds, Heart statling, Diabites, Chart time —are selected and shown in Table 4.

K-means++, IS-DBSCAN, ISB-DBSCAN and C-USB are processed on datasets, with results presented in Table 5.

From Table 5, the application of this algorithm can help achieve better accuracy, entropy evaluation and purity in different dimensions. With the priority of clustering accuracy, the F-measure value of the algorithm is not always as good as other algorithms. As a matter of fact, the clustering accuracy normally outweighs other aspects in real cluster application, thereby, giving priority to clustering accuracy is in line with the actual needs. Based on multiple evaluation criteria, the algorithm prescribed in the paper has been on par with or outperformed the other latest clustering algorithms of the same kind.

4.6. Large datasets

To detect the effect of C-USB on datasets with various scales, several synthesized datasets [55] such as Aggregation, Aggregation+, Flame, Flame+, Sprial, Sprial+ have been selected. The '+' indicates that the points in the dataset have similar distribution and clustering feature with the original dataset. The difference is they are of different scales. The experimental results are shown in Table 6.

Due to the form of expansion, the clustering result indicates that C-USB can process data with various shapes effectively. While the clustering result on k-means++, which is a partition algorithm, cannot process non-spherical clustering problem. Thus, the k-means++ is not partitioned correctly in flame+, making NMI lower and entropy evaluation higher than the other two algorithms. Moreover, due to the partition of internal points and non-internal points, the C-USB can guarantee the clustering effect by adjusting parameters. For instance, in dataset spiral+, C-USB can locate the noise point of the number of clusters correctly by properly enlarging the proportion of dataset's non-internal points when there is a change in the scale of the data; thus, we could reduce difficulties in the process of locating the number of clusters. Still, the algorithm can locate the clusters correctly and guarantee the accuracy. However, the parameters of IS-DBSCAN cannot cope with the enlargement in the scale of data effectively or locate the number of clusters correctly, making the index of entropy reaching 0.6762, which is quite higher than C-USB.

4.7. Analysis on algorithm parameters

Algorithm C-USB involves parameter k and α. Parameter k represents the number of close points in the dataset and α

Please cite this article as: X. Li et al., A clustering algorithm using skewness-based boundary detection, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.023

Table 5
Experimental results and comparison.

Dataset	Algorithm	Accuracy (%)	F-measure (%)	Entropy	Purity
Breast cancer	K-means++	65.52	79.17	0.9076	0.6552
	IS-DBSCAN	71.14	66.79	0.8644	0.7114
	ISB-DBSCAN	97.49	97.24	0.1476	0.9749
	C-USB	98.65	77.28	0.1032	0.9865
seeds	K-means++	89.28	94.34	0.4635	0.8905
	IS-DBSCAN	62.07	75.60	0.8526	0.6355
	ISB-DBSCAN	55.50	68.93	1.1687	0.5550
	C-USB	91.33	86.62	0.4069	0.9133
Heart statling	K-means++	55.56	71.43	0.9906	0.5556
	IS-DBSCAN	56.18	71.65	0.9889	0.5618
	ISB-DBSCAN	57.63	69.46	0.9828	0.5763
	C-USB	83.76	85.19	0.6328	0.8376
Diabetes	K-means++	65.10	78.86	0.9274	0.6510
	IS-DBSCAN	65.28	78.04	0.9298	0.6528
	ISB-DBSCAN	66.48	76.79	0.9188	0.6648
	C-USB	73.02	72.71	0.8296	0.7302
Chart time	K-means++	64.17	78.18	0.8560	0.6417
	IS-DBSCAN	66.67	80.00	0.6661	0.6678
	ISB-DBSCAN	72.18	83.01	0.6118	0.7218
	C-USB	80.55	69.31	0.5143	0.8055

Table 6
Compares the algorithm results with different scales.

Dataset	Number	Evaluation	K-means++	IS-DBSCAN	C-USB
Aggregation	788	Accuracy	0.7652	0.8233	0.9937
		NMI	0.7820	0.8799	0.9837
		Entropy	0.4220	0.4890	0.0303
		Purity	0.8794	0.8265	0.9962
Aggregation+	78,400	Accuracy	0.8126	0.9974	1
		NMI	0.8492	0.9959	1
		Entropy	0.2689	0	0
		Purity	0.9130	1	1
flame	240	Accuracy	0.8375	0.9167	1
		NMI	0.3988	0.7761	0.9188
		Entropy	0.5573	0.0374	0
		Purity	0.8375	0.9957	1
flame+	23,200	Accuracy	0.8588	1	1
		NMI	0.4765	0.9654	0.9654
		Entropy	0.4760	0	0
		Purity	0.8558	1	1
spiral	312	Accuracy	0.3590	1	1
		NMI	0.0001	0.9919	1
		Entropy	1.5835	0	0
		Purity	0.3494	1	1
spiral+	3,120,000	Accuracy	0.3491	0.6635	1
		NMI	0.0003	0.7282	0.8664
		Entropy	1.5842	0.6762	0.0001
		Purity	0.3427	0.6635	1

represents the proportion of the dataset's non-internal points. Breast-cancer dataset is applied to the experiment to clearly state the effect of parameters on algorithms. To facilitate the analysis, as many parameters as possible will be selected and the 3D images in Fig. 9 show the related evaluation indexes. To avoid misunderstanding, please notice the changes in starting values of the axes.

From Fig. 9, the algorithm performance changes along with parameter differences. The value setting of algorithm evaluation develops in the same trend, thereby, values has little to do with algorithm results. The effect of the algorithm would be significantly increased when the value of parameter α is optimum. So, parameter α can influence the effect of the algorithm to a greater extant.

To quantitatively analyze the effect of parameter changing on evaluation of the experiment, the variation tendencies of indexes evaluating the clustering effect of parameter k (160, 200, 240, 280, 320) and α (0.35–0.5) were selected, as shown in Fig. 10.

From Fig. 10, the selection of value of parameter α, which has a greater influence on the effect of the algorithm, is also regular. When the value of parameter α is relatively low, the recall rate of the algorithm is high while the accuracy is low. This is because α represents the proportion of the non-internal point and there might occur the situation where a boundary point is regarded as an internal point when the value of α is relatively low; thus, the noise will be categorized to the cluster and the accuracy will decrease. Instead, too many internal points will be regarded as boundary points when the value of α is relatively high, resulting in a low recall rate. With the precondition that there is a right number of clusters for the algorithm, the selection of parameter α can reflect the different strategies of the users who apply the algorithm. If the application of the dataset requires relatively higher accuracy, the value of the dataset should be properly higher. Equally, if the application requires a higher recall rate, the value of the dataset should be properly lower. As is known, it's impossible to obtain the best accuracy and the most perfect recall rate simultaneously. If the comprehensive analysis on data is required, F-measure should be optimal while securing higher recall rate and accuracy. The adjustment of parameter α reflects the flexibility of algorithm C-USB when confronting different datasets and application scenes.

4.8. Analysis on algorithm time

The time consumption of algorithm can be divided into two parts. The first part is to calculate the boundary of data points, including the time $O(kn^2)$ consumed to calculate the local neighbor points of data points and the time $O(kmn)$ consumed to calculate the boundary degree. The second part is used to find connected matrix $O(mn)$. Therefore, the time complexity is $O(n^2)$. The paper aims to elaborate on skewness-based measurement methods and their corresponding clustering algorithms. That's why no complex index structure is adopted to calculate k nearest neighbor points. Researchers figure out new way to calculate nearest neighbor points for all data points [56–58] within $O(nlogn)$, which can be optimized to be algorithm time complexity by these index structures.

Please cite this article as: X. Li et al., A clustering algorithm using skewness-based boundary detection, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.023

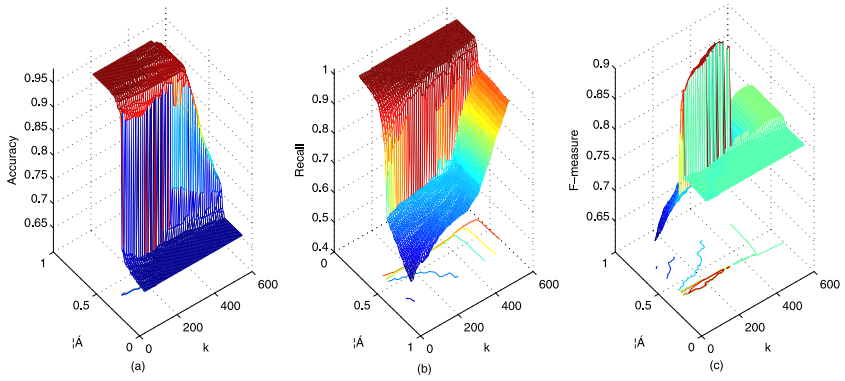

Fig. 9. Effect of the qualitative analysis of parametric variation on clustering result. (a) Accuracy (b) Recall (c) F-measure.

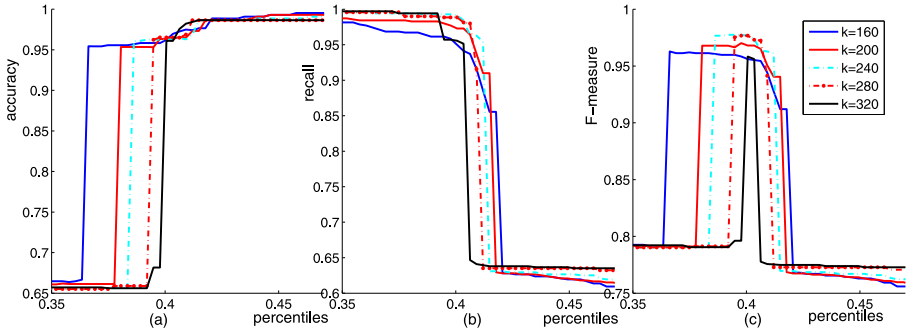

Fig. 10. Effect of the quantitative analysis of parametric variation on clustering result. (a) Accuracy (b) Recall (c) F-measure.

5. Conclusion

Based on the skewed distribution of data points in the boundary area, this paper has proposed a skewness-based measurement method. This method can effectively measure the boundary degree of data points and distinguish the boundary points accurately of the data. The performance of clustering or is on par with the existing latest algorithms. In addition, the boundary detected by the algorithm stated in the paper can better represent the actual situation of clustering boundaries, which playing significant role in data mining.

References

[1] Y. Yang, Z. Ma, Y. Yang, F. Nie, H.T. Shen, Multitask spectral clustering by exploring intertask correlation, IEEE Trans. Cybern 45 (5) (2015) 1083–1094, doi:10.1109/TCYB.2014.2344015.
[2] H. Mashayekhi, J. Habibi, T. Khalafbeigi, S. Voulgaris, M. van Steen, Gdcluster: a general decentralized clustering algorithm, IEEE Trans. Knowl. Data Eng 27 (7) (2015) 1892–1905, doi:10.1109/TKDE.2015.2391123.
[3] P. Shen, C. Li, Distributed information theoretic clustering, IEEE Trans. Signal Process 62 (13) (2014) 3442–3453, doi:10.1109/TSP.2014.2327010.
[4] M. Hahsler, M. Bolanos, Clustering data streams based on shared density between micro-clusters, IEEE Trans. Knowl. Data Eng 28 (6) (2016) 1449–1461, doi:10.1109/TKDE.2016.2522412.
[5] B. Han, L. Liu, E. Omiecinski, Road-network aware trajectory clustering: integrating locality, flow, and density, IEEE Trans. Mobile Comput 14 (2) (2015) 416–429, doi:10.1109/TMC.2013.119.
[6] G. Wang, Q. Song, Automatic clustering via outward statistical testing on density metrics, IEEE Trans. Knowl. Data Eng 28 (8) (2016) 1971–1985, doi:10.1109/TKDE.2016.2535209.
[7] C.F. Tsai, S.C. Huang, An effective and efficient grid-based data clustering algorithm using intuitive neighbor relationship for data mining, in: International Conference on Machine Learning and Cybernetics (ICMLC) 2015, 2, 2015, pp. 478–483, doi:10.1109/ICMLC.2015.7340603.
[8] M.H. Tekieh, B. Raahemi, Importance of data mining in healthcare: a survey, in: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2015, pp. 1057–1062, doi:10.1145/2808797.2809367.
[9] L. Xu, C. Jiang, J. Wang, J. Yuan, Y. Ren, Information security in big data: privacy and data mining, IEEE Access 2 (2014) 1149–1176, doi:10.1109/ACCESS.2014.2362522.
[10] M. Kurras, S. Fahse, L. Thiele, Density based user clustering for wireless massive connectivity enabling internet of things, in: 2015 IEEE Globecom Workshops (GC Wkshps), 2015, pp. 1–6, doi:10.1109/GLOCOMW.2015.7413990.
[11] D. Talia, Making knowledge discovery services scalable on clouds for big data mining, in: 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM) 2015, 2015, pp. 1–4, doi:10.1109/ICSDM.2015.7298015.
[12] J. Zhang, J. Kerekes, An adaptive density-based model for extracting surface returns from photon-counting laser altimeter data, IEEE Geosci. Remote Sens. Lett 12 (4) (2015) 726–730, doi:10.1109/LGRS.2014.2360367.
[13] C.K.S. Leung, R.K. MacKinnon, F. Jiang, Reducing the search space for big data mining for interesting patterns from uncertain data, in: IEEE International Congress on Big Data (BigData Congress) 2014, 2014, pp. 315–322, doi:10.1109/BigData.Congress.2014.53.
[14] R. Shang, P. Tian, L. Jiao, R. Stolkin, A spatial fuzzy clustering algorithm with kernel metric based on immune clone for sar image segmentation, IEEE J. Select. Top. Appl. Earth Observ. Remote Sens 9 (4) (2016) 1640–1652.
[15] R. Shang, Z. Zhang, L. Jiao, C. Liu, Y. Li, Self-representation based dual-graph regularized feature selection clustering, Neurocomputing 171 (2016) 1242–1253, doi:10.1016/j.neucom.2015.07.068.
[16] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, 96, 1996, pp. 226–231.
[17] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, A. Pulvirenti, Enhancing density-based clustering: Parameter reduction and outlier detection, Inf. Syst 38 (3) (2013) 317–330, doi:10.1016/j.is.2012.09.001.
[18] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, An efficient and scalable density-based clustering algorithm for datasets with complex structures, Neurocomputing 171 (2016) 9–22, doi:10.1016/j.neucom.2015.05.109.
[19] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, Lof: identifying density-based local outliers, in: ACM Sigmod Record, 29, ACM, 2000, pp. 93–104.
[20] L. Duan, L. Xu, F. Guo, J. Lee, B. Yan, A local-density based spatial clustering algorithm with noise, Inf. Syst 32 (7) (2007) 978–986.
[21] A.M. Bakr, N.M. Ghanem, M.A. Ismail, Efficient incremental density-based algorithm for clustering large datasets, Alex. Eng. J 54 (4) (2015) 1147–1154, doi:10.1016/j.aej.2015.08.009.
[22] M. Berger, I. Rigoutsos, An algorithm for point clustering and grid generation, IEEE Trans. Syst., Man and Cybern 21 (5) (1991) 1278–1286, doi:10.1109/21.120081.
[23] Y. Wang, M. Lees, W. Cai, Grid-based partitioning for large-scale distributed agent-based crowd simulation, in: Proceedings of the 2012 Winter Simulation Conference (WSC), 2012, pp. 1–12, doi:10.1109/WSC.2012.6465161.
[24] L.J. Du, H.J. Ju, Node clustering algorithm in mobile grid, in: IET International Conference on Information and Communications Technologies (IETICT 2013), 2013, pp. 604–608, doi:10.1049/cp.2013.0108.

Please cite this article as: X. Li et al., A clustering algorithm using skewness-based boundary detection, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.023

[25] G. Divéki, C. Imreh, Grid based online algorithms for clustering problems, in: IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI) 2014, 2014, pp. 159–162, doi:10.1109/CINTI.2014.7028668.
[26] D. Waltz, A theory of the learnable, Commun. ACM 27 (11) (1984).
[27] F. Laviolette, M. Marchand, Pac-bayes risk bounds for stochastic averages and majority votes of sample-compressed classifiers, J. Mach. Learn. Res 8 (7) (2007).
[28] M. Higgs, J. Shawe-Taylor, A pac-bayes bound for tailored density estimation, in: Algorithmic Learning Theory, Springer, 2010, pp. 148–162.
[29] G. Lever, F. Laviolette, J. Shawe-Taylor, Distribution-dependent pac-bayes priors, in: Algorithmic Learning Theory, Springer, 2010, pp. 119–133.
[30] L. Ralaivola, M. Szafranski, G. Stempfel, Chromatic pac-bayes bounds for non-iid data: applications to ranking and stationary β-mixing processes, J. Mach. Learn. Res 11 (2010) 1927–1956.
[31] E. Parrado-Hernández, A. Ambroladze, J. Shawe-Taylor, S. Sun, Pac-bayes bounds with data dependent priors, J. Mach. Learn. Res 13 (1) (2012) 3507–3531.
[32] J. Xu, Y.Y. Tang, B. Zou, Z. Xu, L. Li, Y. Lu, The generalization ability of online SVM classification based on Markov sampling, IEEE Trans. Neural Netw. Learn. Syst 26 (3) (2015) 628–639, doi:10.1109/TNNLS.2014.2361026.
[33] L. Oneto, A. Ghio, S. Ridella, D. Anguita, Fully empirical and data-dependent stability-based bounds, IEEE Trans. Cybern 45 (9) (2015) 1913–1926, doi:10.1109/TCYB.2014.2361857.
[34] A. Behl, P. Mohapatra, C.V. Jawahar, M.P. Kumar, Optimizing average precision using weakly supervised data 37 (12) (2015) 2545–2557, doi:10.1109/TPAMI.2015.2414435.
[35] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM Sigkdd Explor. Newsl 6 (1) (2004) 20–29.
[36] D. Geebelen, J.A.K. Suykens, J. Vandewalle, Reducing the number of support vectors of SVM classifiers using the smoothed separable case approximation, IEEE Trans. Neural Netw. Learn. Syst 23 (4) (2012) 682–688, doi:10.1109/TNNLS.2012.2186314.
[37] J. Manikandan, B. Venkataramani, Evaluation of multiclass support vector machine classifiers using optimum threshold-based pruning technique, IET Signal Process 5 (5) (2011) 506–513, doi:10.1049/iet-spr.2010.0311.
[38] S. Keerthi, Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms, IEEE Trans. Neural Netw 13 (5) (2002) 1225–1229, doi:10.1109/TNN.2002.1031955.
[39] Q.-A. Tran, Q.-L. Zhang, X. Li, Reduce the number of support vectors by using clustering techniques, in: 2003 International Conference on Machine Learning and Cybernetics, 2, 2003, pp. 1245–1248, doi:10.1109/ICMLC.2003.1259678.
[40] C. Xia, W. Hsu, M.L. Lee, B.C. Ooi, Border: efficient computation of boundary points, IEEE Trans. Knowl. Data Eng 18 (3) (2006) 289–303.
[41] L.X. Xue, B.Z. Qiu, Boundary points detection algorithm based on coefficient of variation, Pattern Recogn. Artif. Intell 22 (5) (2009) 799–802.
[42] B.Z. Qiu, F. Yue, J.Y. Shen, Brim: An efficient boundary points detecting algorithm, in: Advances in Knowledge Discovery and Data Mining, Springer, 2007, pp. 761–768.
[43] D. Zhang, H. Tian, Y. Sang, Y. Li, Y. Wu, J. Wu, H. Shen, A clustering algorithm based on density-grid for stream data, in: 13th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT) 2012, 2012, pp. 398–403, doi:10.1109/PDCAT.2012.13.
[44] W. Hu, M. Cheng, G. Wu, L. Wu, Research on parallel data stream clustering algorithm based on grid and density, in: International Conference on Computer Science and Mechanical Automation (CSMA) 2015, 2015, pp. 70–75, doi:10.1109/CSMA.2015.21.
[45] R. Vallakati, A. Mukherjee, P. Ranganathan, A density based clustering scheme for situational awareness in a smart-grid, in: IEEE International Conference on Electro/Information Technology (EIT) 2015, 2015, pp. 346–350, doi:10.1109/EIT.2015.7293366.
[46] S.Y. Xia, Z.Y. Xiong, Y. He, Relative density-based classification noise detection, Optik 125 (2014) 6829–6834.
[47] T. Drugman, Residual excitation skewness for automatic speech polarity detection, IEEE Signal Process. Lett 20 (4) (2013) 387–390, doi:10.1109/LSP.2013.2249661.
[48] X. Geng, L. Meng, L. Li, L. Ji, K. Sun, Momentum principal skewness analysis, IEEE Geosci. Remote Sens. Lett 12 (11) (2015) 2262–2266, doi:10.1109/LGRS.2015.2465814.
[49] G. Georgiou, K. Voigt, Stochastic computation of moments, mean, variance, skewness and kurtosis, Electron. Lett 51 (9) (2015) 673–674, doi:10.1049/el.2015.0066.
[50] G. Karypis, E.H. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, Computer 32 (8) (1999) 68–75.
[51] Z. Yu, P. Luo, J. You, H.S. Wong, H. Leung, S. Wu, J. Zhang, G. Han, Incremental semi-supervised clustering ensemble for high dimensional data clustering, IEEE Trans. Knowl. Data Eng 28 (3) (2016) 701–714, doi:10.1109/TKDE.2015.2499200.
[52] J. Kim, D. Han, Y.W. Tai, J. Kim, Salient region detection via high-dimensional color transform and local spatial support, IEEE Trans. Image Process 25 (1) (2016) 9–23, doi:10.1109/TIP.2015.2495122.
[53] L. Jiao, F. Shang, F. Wang, Y. Liu, Fast semi-supervised clustering with enhanced spectral embedding, Pattern Recogn 45 (12) (2012) 4358–4369, doi:10.1016/j.patcog.2012.05.007.
[54] R. Shang, Z. Zhang, L. Jiao, W. Wang, S. Yang, Global discriminative-based nonnegative spectral clustering, Pattern Recogn 55 (2016) 172–182, doi:10.1016/j.patcog.2016.01.035.
[55] synthesized datasets, http://cs.uef.fi/sipu/datasets/.
[56] X. Liu, C. Deng, B. Lang, D. Tao, X. Li, Query-adaptive reciprocal hash tables for nearest neighbor search, IEEE Trans. Image Process 25 (2) (2016) 907–919, doi:10.1109/TIP.2015.2505180.
[57] S. He, Q. Yang, R.W.H. Lau, M.H. Yang, Fast weighted histograms for bilateralfiltering and nearest neighbor searching, IEEE Trans. Circuits Syst. Video Technol 26 (5) (2016) 891–902, doi:10.1109/TCSVT.2015.2430671.
[58] S. guang Liu, Y. wei Wei, Fast nearest neighbor searching based on improved vp-tree, Pattern Recogn. Lett 6061 (2015) 8–15, doi:10.1016/j.patrec.2015.03.017.

Xiangli Li an Associate 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.

Qiong Han received his B.S. degree in July 2014 and currently is a M.S. candidate majored in Computer Science all in School of Information Engineering, Zhengzhou University. His research interests include Pattern Recognition and Data Mining.

Baozhi Qiu received his Ph.D. degree in computer science from XianJiaotong University in 2006. He is currently a full Professor in School of Information Engineering, Zhengzhou University, China. His research interests include data mining, database performance issues and advanced applications.

Please cite this article as: X. Li et al., A clustering algorithm using skewness-based boundary detection, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.023