CAAI Transactions on Intelligence Technology

Research Article

Two-phase clustering algorithm with density exploring distance measure

ISSN 2468-2322
Received on 3rd January 2018
Revised on 11th February 2018
Accepted on 20th February 2018
doi: 10.1049/trit.2018.0006
www.ietdl.org

Jingjing Ma, Xiangming Jiang, Maoguo Gong ✉

Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, PO Box 224, Xi'an 710071, People's Republic of China
✉ *E-mail: gong@ieee.org*

Abstract: Here, the authors propose a novel two-phase clustering algorithm with a density exploring distance (DED) measure. In the first phase, the fast global *K*-means clustering algorithm is used to obtain the cluster number and the prototypes. Then, the prototypes of all these clusters and representatives of points belonging to these clusters are regarded as the input data set of the second phase. Afterwards, all the prototypes are clustered according to a DED measure which makes data points locating in the same structure to possess high similarity with each other. In experimental studies, the authors test the proposed algorithm on seven artificial as well as seven *UCI* data sets. The results demonstrate that the proposed algorithm is flexible to different data distributions and has a stronger ability in clustering data sets with complex non-convex distribution when compared with the comparison algorithms.

1 Introduction

When data available are unlabelled, the classification problems are usually referred to as the unsupervised classification, or clustering. In clustering, a set of patterns, usually vectors in a multi-dimensional space, are grouped into clusters in such a way that patterns in the same cluster are similar in some sense and patterns in different clusters are dissimilar [1]. Many clustering approaches, such as the *K*-means algorithm (KM) [2], partition a data set into a specified number of clusters by minimising certain criteria. Therefore, they can be treated as one optimisation problem. Since various successful algorithms such as *K*-means [2], STING [3], CLIQUE [4], and CURE [5] have been proposed in recent years, clustering has become a common and important technique for statistical data analysis, which is widely used in many fields including machine learning, data mining [6], pattern recognition [7, 8], and image analysis [9].

Choosing proper dissimilarity measures is one of the key points in designing of clustering algorithm [10–13]. Euclidean distance is a traditional one which is widely applied. Clustering methods such as the KM with the Euclidean distance are able to achieve satisfactory performance on data sets with compact spherical distributions, but tend to fail on data sets organised in other complex shapes [14, 15]. Therefore, it is necessary for researchers to design more flexible measurement [16, 17]. Su and Chou [18] proposed a non-metric measure based on the concept of point symmetry. According to this measure, a symmetry-based version of the KM has been given. This algorithm assigns data samples to the same cluster centre as if they present a symmetrical structure with respect to that centre. Recently, Charalampidis [19] developed a dissimilarity measure for directional patterns represented by rotation-variant vectors and further introduced a circular KM to cluster vectors containing directional information.

In [20, 21], we designed a density-sensitive dissimilarity measure for the KM and the evolutionary clustering algorithm, respectively. This measure, termed density exploring distance (DED), makes data sets with complex structural features of distribution to be better reflected. As introduced in [20, 21], the algorithms perform well on data sets with a complex distribution. However, this measure has a major disadvantage that it costs too much time in searching for the shortest path between any two data samples. When the size of data set increases, this drawback becomes extremely obvious that it will not be suitable for large-scale clustering problems any longer. To solve this problem, in this study, we distinguish our proposed method in two phases. In the first phase, the fast global *K*-means clustering algorithm [22] is utilised to acquire some subsets of data with spherical distributions whose centres can precisely represent the property of the distribution of the input data set. Regarding the centres of these subsets as the input data set, the complexity for calculating DED of any two samples can be reduced significantly. In the second phase, we adopt the *K*-means clustering with DED measure. The proposed method can perform well on data sets organised in any shapes without taking too much time. Experimental studies on seven artificial data sets as well as seven UCI data sets show that this novel algorithm is suitable for identifying data with complex non-convex distributions compared with the fast global *K*-means algorithm (FGKM) [22], the genetic algorithm-based clustering (GAC) algorithm [23], the DED-based *K*-means algorithm [20], the density-sensitive evolutionary clustering (DSEC) [21], and the KM [2].

In the following sections, we start with a brief description on the main steps of the proposed method and our motivation in Section 2. The proposed approach is introduced in detail in Section 3. Section 4 provides experimental results and comparisons. Finally, concluding remarks are summarised in Section 5.

2 Motivation

Clustering is a process that divides a data set into clusters where points are similar in the same cluster. Dissimilarity measure plays an important role in promoting high performance of a clustering method [24–26]. Also, the computational complexity is also concerned to evaluate the effectiveness of a clustering approach. Taking into account the issues mentioned above, the proposed method is made up of two main phases: (i) obtaining initial clusters with FGKM; (ii) embedding DED measure into KM for precise clustering.

2.1 *Motivation of adopting the DED measure*

As aforementioned in Section 1, traditional KM with Euclidean distance works well on data sets with compact spherical

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

distribution, but fails on data sets with other complex distribution. To solve this problem, we introduce the DED measure below.

This measure was firstly described in work [20]. Through a large amount of observations, two consistency characters of data clustering have been found, which are coincident with the prior assumption of consistency in semi-supervised learning [27–31]. The so-called local consistency refers to that data samples close in location possess high affinity with each other, while global consistency refers to that data samples locating in the same structure possess high affinity with each other.

For many real-world problems, the distribution of data samples generally takes different complex structures in Euclidian feature space. However, the classical Euclidian distance metric can only reflects the local consistency, while fails to describe the global consistency. Fig. 1 gives an illustration on this problem. The affinity between point a and point e is expected to be higher than that between point a and point f. In other words, we are looking for a distance measure which is able to identify that point a is closer to point e than to point f. However, point a is much closer to point f than point e in terms of Euclidian distance metric. Hence, Euclidean distance metric is not suitable for many complicated real-world problems.

Based on many observations, we can find that the density distribution of a data set reflects both the local and global consistency well in many conditions. As shown in Fig. 1, data points in the same cluster tend to lie in a region of high density. Therefore, we design a data-dependent distance measure in terms of the character of data density which can reflect both the local and the global consistency. That is to say the proposed method elongates the distance between points in different density parts and shortens that in the same part.

2.2 Motivation of adopting two phases

As mentioned above, another problem arises followed by a satisfactory performance when DED measure is introduced. Since calculating the shortest path between any two points is a time-consuming procedure, it is not suitable for large-scale data sets to utilise DED measure.

If only using the information of some points from the entire data set can acquire the correct clustering result, the computational complexity must reduce greatly. Based on many observations, we can find that it is not essential for every point to reflect the distribution of the data set. As shown in Fig. 2, we can ignore some redundant information of part of the data set, and construct a smaller data set (as shown in Fig. 2a) to represent the distribution of the original data set (as shown in Fig. 2b). That is to say, dealing with these data points can also obtain better clustering result. Meanwhile, we discover that a data set, whether it is with compact spherical distribution or complex distribution, can be divided into several subclusters with spherical distribution. This process can be easily achieved by selecting an appropriate number of clusters and using K-means method for clustering. The subcluster centres of the data set are good representatives of the distribution character of the entire data set. Then dealing with these representative set of points by DED-based method cannot only achieve satisfactory result but also greatly reduce computational complexity. Based on these ideas above, the proposed method is divided into two phases, which will be described in detail in the following section.

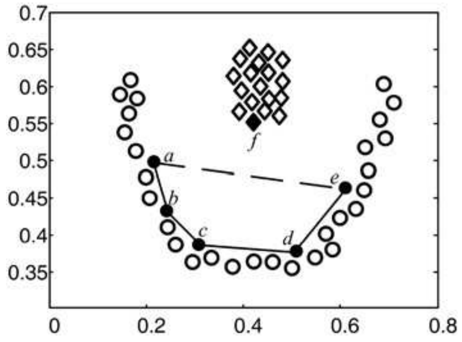

Fig. 1 *Illustration on why Euclidian distance metric cannot reflect the global consistency*

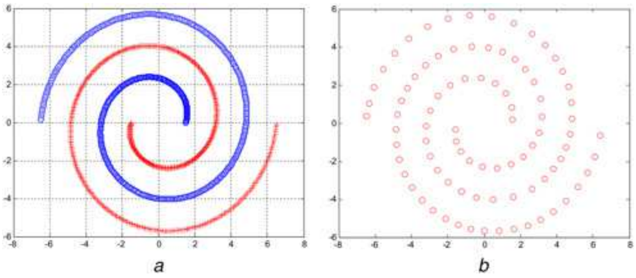

Fig. 2 *Illustration on representative set of points after the first phase*

a Original data set
b Simplified representatives of the original data set

3 Methodology

In this section, we describe the proposed two-phase clustering algorithm with density exploring distance measure (TPCDED) in detail. Its framework is summarised as follows.

Algorithm 1: Two-phase clustering algorithm with density exploring distance measure

```
Input: Data set $\{x_i\}_{i=1}^{n}$; cluster number $K$; maximum iteration number $t_{\max}$; stop threshold $e$.
Output: Partition of the data set $C_1, C_2, \ldots, C_k$.
Step 1 Cluster the data set $\{x_i\}_{i=1}^{n}$ and adaptively determine the number of clusters $K'$ by using the FGKM (as described in Section 3.1), and set $\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_{K'}\}$.
Step 2 For any two points $\mu_i$, $\mu_j$, compute the DED between them (as described in Section 3.2). Store them for further use. Random select $K$ points from $\boldsymbol{\mu}$ as the initial prototypes of the section-phase clustering.
Step 3 Each point is assigned to the cluster with the minimum DED from its prototype to the point.
Step 4 Recalculate the prototype of each cluster.
Step 5 If no point changes its category or the number of iterations reaches the maximum number $t_{\max}$, stop; otherwise, go to step 3.
```

Step 2 is designed for speeding up the whole procedure. The most time-consuming part in our algorithm is the calculation of DED, so we try to avoid this computation during the update procedure. All the distances between any two points are calculated and stored firstly. We abandon the original K-means method which calculates the mean value vectors as cluster representatives in step 4. This is because mean values cannot represent density judged areas properly. Instead, for each point in current cluster, we calculate the sum of its DEDs to all the other points in the same cluster, and then choose the point which maximises the sum of DEDs as the new prototype of this cluster. In this way, all the prototypes are chosen from $\boldsymbol{\mu}$ in step 1, and we only need to call the distances stored in step 2.

3.1 Selecting initial prototypes

We apply the FGKM [22] for the first-phase clustering. This method increases the number of clusters iteratively until an appropriate value which can be determined adaptively by selecting a certain inflection point of the error curve.

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

To solve a clustering problem with k clusters, the method works as follows. It starts with one cluster ($k = 1$) and finds its optimal centre which minimises the clustering error. Then, it solves the problem with two clusters ($k = 2$) by computing an upper bound $E_n \le E - b_n$ on the resulting error E_n for all possible allocation positions x_n, where E is the clustering error in the one-cluster problem and b_n is defined as

$$b_n = \sum_{i=1}^{N} \max\left(d_{k-1}^{i} - \|x_n - x_i\|^2, 0\right) \qquad (1)$$

where d_{k-1}^{i} is the squared distance between x_i and the closest centre among the $k - 1$ cluster centres obtained so far (i.e. centre of the cluster where x_i belongs). The quantity b_n measures the guaranteed reduction obtained by inserting a new cluster centre at position x_n. Then, the new cluster centre is initialised at point x_j, ($j = \arg\max_n \{b_n\}$) which minimises E_n (or equivalently maximises b_n). Suppose the solution of the $(k-1)$-cluster problem is $(m_1^*(k-1), (m_2^*(k-1), \ldots, (m_{k-1}^*(k-1))$ and a new cluster centre is inserted at location x_n. Then the new centre will allocate all points x_i whose squared distance from x_n is smaller than d_{k-1}^{i}. Therefore, the clustering error for each x_i will decrease by $d_{k-1}^{i} - \|x_n - x_i\|^2$. The summation over all x_i provides the quantity b_n for a specific insert location x_n. By minimising the clustering error step by step, k initial centres can be obtained for the k-cluster problem.

There is a key problem that how many clusters should be divided into the first phase. Too many clusters will burden extra computational complexity on the proposed algorithm, but when the number of clusters is too small, the representative set of points cannot reflect the distribution of original data set well. The FGKM select cluster centres increasingly according to a strict criterion, so that these cluster centres can approximate the results of hierarchical clustering algorithm. This character helps us select appropriate number of clusters by selecting the inflection point of the error curve, which is very flexible and embodied in how to select the error curve and find the inflection point. The inflection point is generalised to a special point satisfying certain conditions. In this paper, we choose the within-cluster sum of squares of all clusters to be the error curve with an increasing number of clusters, and the first point satisfying the criterion that the difference between each of three successive points on the error curve is below a certain threshold is selected to be the knee point. The number of clusters corresponding to the knee point is the number needed to be determined in the first phase.

3.2 Density exploring distance

In the second phase, we adopt DED measure as dissimilarity measure in the modified K-means method. A detailed description of DED measure is given as below.

In our method, the whole data set is modelled as a weighted undirected graph $G = (V, E)$. Data points are taken as the nodes V. Edges $E = W_{i,j}$ are weighted by the distance between points x_i and x_j. We expect the distance measure assigns a high affinity between two points if they can be linked by a path running within a region of high density, and a low affinity otherwise. In other words, this measure should elongate the paths that cross low-density regions, and simultaneously shorten those only cross high-density regions. In the illustration example in Fig. 1, that is, we look for a measure of distance, according to which point a is closer to point e than to point f as mentioned above.

To formalise this intuitive notion of dissimilarity, we firstly define a so-called sensitised distance. Different from traditional point of view, a distance measure describing the global consistency does not always satisfy the triangle inequality under the Euclidean metric. In other words, a direct connected path between two points should not always be the shortest one. As shown in Fig. 1, global consistency requires that the solid line path is shorter than the straight dashed path, i.e. $\overline{ab} + \overline{bc} + \overline{cd} + \overline{de} < \overline{ae}$. Enlightened by this property, we define the sensitised distance as follows.

Definition 1: The sensitised distance $\mathrm{SD}(x_i, x_j)$ is defined as

$$\mathrm{SD}(x_i, x_j) = \rho^{\mathrm{dist}(x_i, x_j)} - 1 \qquad (2)$$

where $\mathrm{dist}(x_i, x_j)$ is the Euclidean distance between x_i and x_j, and $\rho > 1$ the flexing factor.

The distance sensitivity between two points can be adjusted by the flexing factor ρ. In virtue of the sensitised distance, we further introduce the new distance dissimilarity measure DED, which calculates the distance between a pair of points by searching for the shortest path in the graph.

Definition 2: Let data points be the nodes of graph $G = (V, E)$ and $p \in V^l$ be a path of length $l = |p| - 1$ connecting the nodes p_1 and $p_{|p|}$, in which $(p_k, p_{k+1}) \in E$, $1 \le k \le |p| - 1$. Let P_{ij} denote the set of all paths connecting nodes x_i and x_j. The DED between x_i and x_j is defined as

$$D\left(x_i, x_j\right) = \min_{p \in P_{ij}} \sum_{k=1}^{|p|-1} L\left(p_k, p_{k+1}\right) \qquad (3)$$

We can observe that this measure tries to search data distribution and satisfies the four conditions for a distance metric, i.e. $D(x_i, x_j) = D(x_j, x_i)$; $D(x_i, x_j) \ge 0$; $D(x_i, x_j) \le D(x_i, x_k) + D(x_k, x_j)$ for all x_i, x_j, x_k; and $D(x_i, x_j) = 0$ if and only if $x_i = x_j$.

This distance measure judges any two points in the same dense area, which are connected by a number of short edges within this area, while the linkage between any two points in different dense areas contains much longer edges between these dense areas. Therefore, the distances between data points in different dense areas are elongated and that in the same dense area are shortened simultaneously. As a result, the property of data sets with complex distribution can be effectively described.

4 Experimental results

In order to validate the performance of TPCDED, we design experiments on seven artificial data sets and seven UCI data sets [32]. The information of these artificial and UCI data sets are given in Table 1. The results are compared with a modified KM using the density-sensitive dissimilarity measure (DSKM) [20], the FGKM [22], the GAC technique [23], the DSEC algorithm [21], and the KM [2].

The main drawback of DSKM is that the centre of each cluster is the geometrical centre in each iteration, which discounts its ability of reflecting the global consistency. TPCDED overcomes this drawback by updating the cluster representatives within data set (by selecting the point which maximises the sum of DEDs in each cluster as the cluster representative). Although DSEC can

Table 1 Data sets used for experiments

Data set	Number of samples	Number of features	Number of clusters
Square1	1000	2	4
Square4	1000	2	4
Long1	1000	2	2
Spiral	1000	2	2
Sizes5	1000	2	4
Line-blobs	266	2	3
Sticks	512	2	4
Iris	150	4	4
Wine	178	13	3
Breast	277	9	2
Zoo	101	16	7
German	1000	20	2
Pimaindians	768	8	2
Musk	6598	166	2
Page	5473	10	5

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and
Chongqing University of Technology under the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/3.0/)

perform well on non-convex data sets, it is a time-consuming method. Finally, FGKM and GAC belong to Euclidean distance measure-based approaches.

In all the artificial and UCI problems, the desired clusters' number is set in advance. For TPCDED, DSKM, and KM, $t_{\max}$ is set as 500, and the stop threshold is set as 10^{-4}. The parameter settings for GAC and DSEC are given in Table 2. The sensitivity test of parameter based on the above 14 data sets shows that the results of TPCDED only vary slightly when the flexing factor ρ is set within $(1, e^{18}]$.

Table 2 Parameter settings for GAC and DSEC

Parameter	Value
population size	20
number of generation	500
probability of crossover	0.8
probability of mutation	0.1

4.1 Experimental results on artificial data sets

In this section, we evaluate the performance of TPCDED on seven artificial data sets, i.e. Square1, Square4, Long1, Spiral, Size5, Line-blobs, and Sticks. To show the performance visually, the typical clustering results obtained by TPCDED, FGKM, DSKM, GAC, and KM are shown in Figs. 3 and 4.

Clustering quality is evaluated by percentage of accuracy. We perform 30 independent runs on each problem in order to test TPCDED when compared with DSKM, FGKM, GAC, and KM. The average results of clustering accuracy are shown in Table 3. From Table 3, we can find clearly that TPCDED achieves the best performance in clustering all the seven data sets. DSKM also performs well on Long1 and Spiral data sets, FGKM and KM do the best on Square1 and Square4 data set, too. FGKM, GAC, and KM only obtain satisfactory results for the three spheroid data sets, i.e. Square1, Square4, and Size5. The structures of the other four data sets do not satisfy convex distribution. DSKM can recognise two of the four complex distributions successfully, this

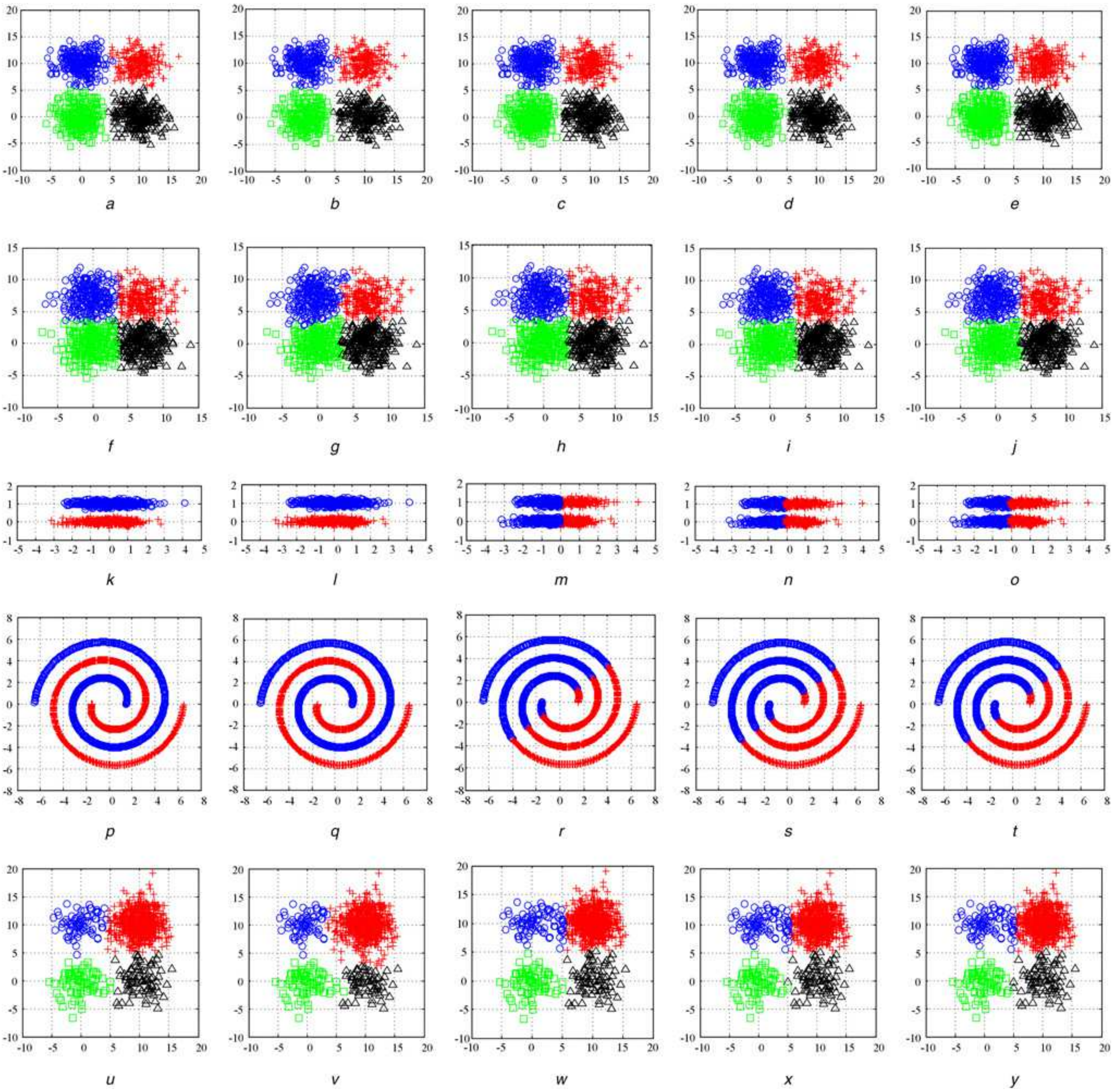

Fig. 3 *Experimental results on typical artificial data sets. The first column to the fifth column are the results obtained by TPCDED, DSKM, FGKM, GAC, and KM, respectively. The top row to the bottom row are the results on Square1, Square4, Long1, Spiral, Sizes5, respectively*

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

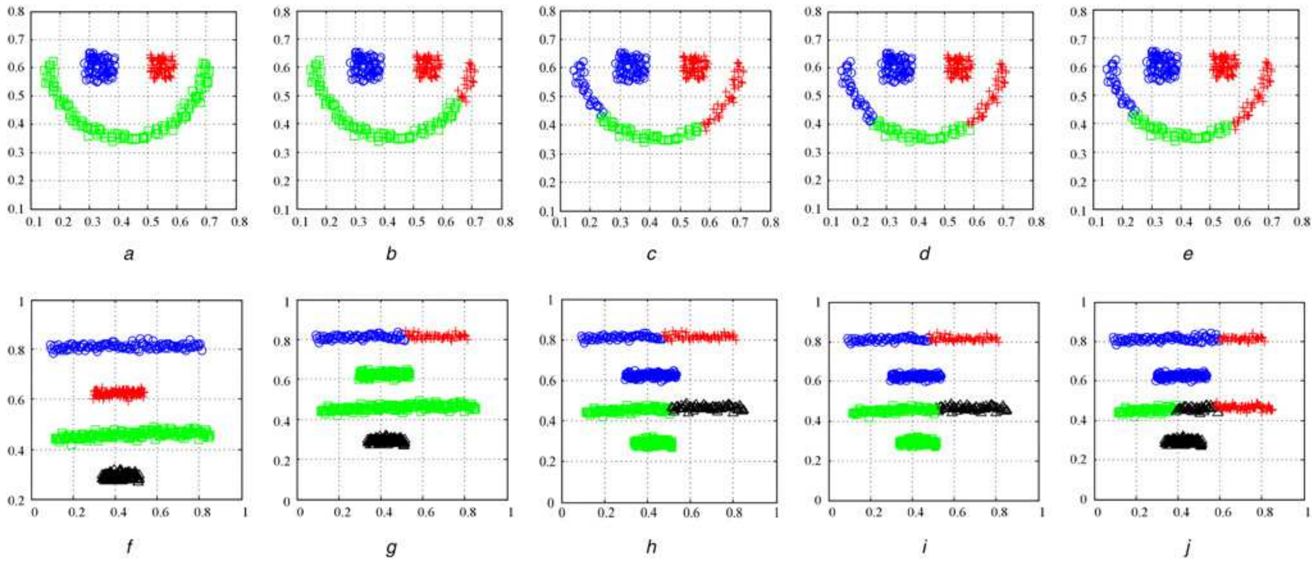

Fig. 4 *Experimental results on typical artificial data sets. The first column to the fifth column are the results obtained by TPCDED, DSKM, FGKM, GAC, and KM, respectively. The top row to the bottom row are the results on Line-blobs and Sticks, respectively*

Table 3 Accuracy results of TPCDED, DSKM, GAC, and KM on artificial data sets

Data sets	Percentage of accuracy				
	TPCDED	DSKM	FGKM	GAC	KM
Square	0.9872	0.8550	**0.9900**	0.9899	**0.9900**
Square4	0.9285	0.8547	**0.9350**	0.9341	**0.9350**
Long1	**1.0000**	**1.0000**	0.5140	0.5620	0.5464
Spiral	**1.0000**	**1.0000**	0.5920	0.5960	0.5927
Sizes5	**0.9838**	0.8657	0.9760	0.9755	0.7744
Line-blobs	**1.0000**	0.9038	0.7444	0.7368	0.7425
Sticks	**1.0000**	0.7628	0.7207	0.7312	0.6895

The bold in this table represents the best results achieved among these algorithms.

Table 4 Accuracy results of TPCDED, DSKM, FGKM, GAC, and KM on UCI data sets

Data sets	Percentage of accuracy				
	TPCDED	DSKM	FGKM	GAC	KM
Iris	0.9077	0.7929	0.8867	0.8996	0.7938
Breast	0.7076	0.5386	0.6065	0.4188	0.4897
Zoo	**0.8614**	0.6455	0.7921	0.7619	0.7007
German	**0.7000**	0.6779	0.5970	0.5864	0.5970
Pimaindians	**0.6510**	0.5635	0.5482	0.5287	0.5482
Musk	**0.8459**	0.8457	0.5132	0.5040	0.5150
Page	**0.8771**	0.8276	0.8012	0.8030	0.7312

The bold in this table represents the best results achieved among these algorithms.

indicates that the manifold distance metric is suitable to measure complicated data distribution. When comparisons are made between TPCDED and DSKM, both of them can obtain the true distribution of Long1 and Spiral data sets in all the 30 runs, but DSKM cannot do it on the Line-blobs and Sticks data sets. Furthermore, the proposed TPCDED performs better than DSKM for clustering Square1, Square4, and Size5 data sets.

4.2 Experimental results on UCI data sets

We also choose seven real-world data sets from the UCI machine learning repository, i.e. Iris, Breast, Zoo, German, Pimaindians, Musk, and Page, to evaluate the performance of TPCDED. Compared with DSKM, FGKM, GAC, and KM, the average clustering accuracy of 30 independent runs of TPCDED on each data set is shown in Table 4.

From Table 4, we can find clearly that TPCDED achieves the best performance on all of the seven clustering problems. For Iris and Zoo data sets, FGKM and GAC achieve better results than DSKM and KM. For Breast data set, only TPCDED can obtain satisfactory result. On the rest data sets, our proposed TPCDED and DSKM perform more effectively than other method.

4.3 Comparisons of computational time

Both DSKM and DSEC have a drawback that they cost too much time when the scale of data set is large. We perform 30 independent runs on each problem to test the time efficiency of DSEC and TPCDED. The average results of clustering accuracy and computational time are shown in Table 5.

Table 5 Time efficiency results of TPCDED and DSEC

Data set	Percentage of accuracy		Time, s	
	TPCDED	DSEC	TPCDED	DSEC
Square1	0.9872	**0.9895**	**7.1883**	14.7406
Square4	0.9285	**0.9336**	**7.3906**	14.9531
Long1	1.0000	1.0000	**6.6599**	14.1313
Spiral	1.0000	1.0000	**6.7021**	13.5844
Sizes5	0.9838	**0.9885**	**7.1906**	15.0156
Line-blobs	1.0000	1.0000	**1.1339**	2.2719
Sticks	1.0000	1.0000	**1.8844**	4.9063
Iris	**0.9077**	0.9013	**0.8036**	1.5562
Breast	0.7076	0.7076	**1.5984**	2.6809
Zoo	**0.8614**	0.7921	**1.2039**	1.6438
German	0.7000	0.7000	**6.9781**	14.1156
Pimaindians	0.6510	0.6510	**5.3156**	11.7094
Musk	**0.8457**	—	**807.1906**	—
Page	**0.8771**	—	**501.3687**	—

The '—' in the table means that the result does not come out within 24 h. The bold in this table represents the best results achieved among these algorithms.

From Table 5, we can see that TPCDED maintains high clustering accuracy and greatly improves the computational efficiency simultaneously. Its computational time is much less than that of DSEC. For Musk and Page, DSEC cannot obtain results within 24 h because of its high computational complexity, while TPCDED achieves satisfactory results within limited time.

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)

5 Concluding remarks

In this paper, we propose a two-phase clustering algorithm with a DED measure. Since the DED measure can identify non-convex clustering structures, the proposed algorithm can achieve satisfactory performances on data sets with complex distributions. A fast global prototype selection strategy is applied to find global optimum clustering solutions and make original data sets to be represented by some centres of clusters. The number of representatives is determined adaptively. This procedure greatly speeds up the whole algorithm. This method maintains the accuracy of clustering and saves a lot of time simultaneously.

Experimental results on seven artificial as well as seven UCI data sets show that the proposed algorithm is flexible to different data distributions and has stronger ability to identify complex non-convex clusters than the compared algorithms in terms of cluster quality and computational time.

6 References

[1] Jain, A., Murty, M., Flynn, P.: 'Data clustering: a review', *ACM Comput. Surv.*, 1999, **31**, (3), pp. 264–323

[2] Hartigan, J.A., Wong, M.A.: 'Algorithm as 136: a *k*-means clustering algorithm', *J. R. Stat. Soc. C, Appl. Stat.*, 1979, **28**, (1), pp. 100–108

[3] Wang, W., Yang, J., Muntz, R., *et al.*: 'STING: a statistical information grid approach to spatial data mining'. Proc. of the 23rd Int. Conf. on Very Large Data Bases, Athens, Greece, 1997, pp. 186–195

[4] Agrawal, R., Gehrke, J., Gunopulos, D., *et al.*: 'Automatic subspace clustering of high dimensional data for data mining applications'. Proc. of ACM-SIGMOND Int. Conf. Management on Data, Seattle, Washington, USA, 1998, pp. 94–105

[5] Guha, S., Rastogi, R., Shim, K.: 'CURE: an efficient clustering algorithm for large databases'. Proc. of ACM-SIGMOND Int. Conf. Management on Data, Seattle, Washington, USA, 1998, pp. 73–84

[6] Wu, J., Wu, Z., Cao, J., *et al.*: 'Fuzzy consensus clustering with applications on big data', *IEEE Trans. Fuzzy Syst.*, 2017, **25**, (6), pp. 1430–1445

[7] Huang, J., Liu, J.: 'A similarity-based modularization quality measure for software module clustering problems', *Inf. Sci.*, 2016, **342**, pp. 96–110

[8] Xia, G., Sun, H., Feng, L., *et al.*: 'Human motion segmentation via robust kernel sparse subspace clustering', *IEEE Trans. Image Process.*, 2018, **27**, (1), pp. 135–150

[9] Meng, F., Li, H., Wu, Q., *et al.*: 'Globally measuring the similarity of superpixels by binary edge maps for superpixel clustering', *IEEE Trans. Circuits Syst. Video Technol.*, 2016, doi: 10.1109/TCSVT.2016.2632148

[10] Taşdemir, K., Yalçin, B., Yildirim, I.: 'Approximate spectral clustering with utilized similarity information using geodesic based hybrid distance measures', *Pattern Recognit.*, 2015, **48**, (4), pp. 1465–1477

[11] Mori, U., Mendiburu, A., Lozano, J.A.: 'Similarity measure selection for clustering time series databases', *IEEE Trans. Knowl. Data Eng.*, 2016, **28**, (1), pp. 181–195

[12] Ye, J.: 'Single-valued neutrosophic clustering algorithms based on similarity measures', *J. Classif.*, 2017, **34**, (1), pp. 148–162

[13] Son, L.H.: 'Generalized picture distance measure and applications to picture fuzzy clustering', *Appl. Soft Comput.*, 2016, **46**, (C), pp. 284–295

[14] Cao, F., Liang, J., Li, D., *et al.*: 'A dissimilarity measure for the *k*-modes clustering algorithm', *Knowl.-Based Syst.*, 2012, **26**, pp. 120–127

[15] Yang, P., Zhu, Q., Huang, B.: 'Spectral clustering with density sensitive similarity function', *Knowl.-Based Syst.*, 2011, **24**, (5), pp. 621–628

[16] Ferrari, D.G., De Castro, L.N.: 'Clustering algorithm selection by meta-learning systems: a new distance-based problem characterization and ranking combination methods', *Inf. Sci.*, 2015, **301**, pp. 181–194

[17] Lu, Y., Hou, X., Chen, X.: 'A novel travel-time based similarity measure for hierarchical clustering', *Neurocomputing*, 2016, **173**, pp. 3–8

[18] Su, M.-C., Chou, C.-H.: 'A modified version of the *k*-means algorithm with a distance based on cluster symmetry', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001, **23**, (6), pp. 674–680

[19] Charalampidis, D.: 'A modified *k*-means algorithm for circular invariant clustering', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, **27**, (12), pp. 1856–1865

[20] Wang, L., Bo, L., Jiao, L.: 'A modified *k*-means clustering with a density-sensitive distance metric'. Proc. of the Int. Conf. on Rough Sets and Knowledge Technology, Chongquing, China, 2006, pp. 544–551

[21] Gong, M., Jiao, L., Wang, L., *et al.*: 'Density-sensitive evolutionary clustering'. Proc. of the 11th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, Nanjing, China, 2007, pp. 507–514

[22] Likas, A., Vlassis, N., Verbeek, J.J.: 'The global *k*-means clustering algorithm', *Pattern Recognit.*, 2003, **36**, (2), pp. 451–461

[23] Maulik, U., Bandyopadhyay, S.: 'Genetic algorithm-based clustering technique', *Pattern Recognit.*, 2000, **33**, (9), pp. 1455–1465

[24] Kang, Q., Liu, S., Zhou, M., *et al.*: 'A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence', *Knowl.-Based Syst.*, 2016, **104**, pp. 156–164

[25] Kang, Z., Peng, C., Cheng, Q.: 'Twin learning for similarity and clustering: a unified kernel approach'. AAAI, San Francisco, California, USA, 2017, pp. 2080–2086

[26] Machné, R., Murray, D.B., Stadler, P.F.: 'Similarity-based segmentation of multi-dimensional signals', *Sci. Rep.*, 2017, **7**, (1), p. 12355

[27] Zhou, D., Bousquet, O., Lal, T.N., *et al.*: 'Learning with local and global consistency'. Advances in Neural Information Processing Systems, Vancouver, Canada, 2004, pp. 321–328

[28] Zong, Y., Xu, G., Zhang, Y., *et al.*: 'A robust iterative refinement clustering algorithm with smoothing search space', *Knowl.-Based Syst.*, 2010, **23**, (5), pp. 389–396

[29] Zhu, S., Wang, D., Li, T.: 'Data clustering with size constraints', *Knowl.-Based Syst.*, 2010, **23**, (8), pp. 883–889

[30] Bousquet, O., Chapelle, O., Hein, M.: 'Measure based regularization'. Advances in Neural Information Processing Systems, Vancouver, Canada, 2004, pp. 1221–1228

[31] Blum, A., Chawla, S.: 'Learning from labeled and unlabeled data using graph mincuts'. Proc. of the 18th Int. Conf. on Machine Learning (ICML), Williamstown, MA, USA, 2001, pp. 19–26

[32] Blake, C.L., Merz, C.J.: 'UCI repository of machine learning databases'. Technical Report, Department of Information and Computer Science, University of California, Irvine, CA, 1998

This is an open access article published by the IET, Chinese Association for Artificial Intelligence and Chongqing University of Technology under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/)