Pattern Recognition Letters 97 (2017) 46–53

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

A novel density peaks clustering algorithm for mixed data

Mingjing Du[a], Shifei Ding[a,b,*], Yu Xue[c]

[a] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, 221116, China
[b] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
[c] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China

ARTICLE INFO

Article history:
Received 31 May 2016
Available online 3 July 2017

Keywords:
Data clustering
Density peaks
Entropy
Mixed data

ABSTRACT

The density peaks clustering (DPC) algorithm is well known for its power on non-spherical distribution data sets. However, it works only on numerical values. This prohibits it from being used to cluster real world data containing categorical values and numerical values. Traditional clustering algorithms for mixed data use a pre-processing based on binary encoding. But such methods destruct the original structure of categorical attributes. Other solutions based on simple matching, such as K-Prototypes, need a user-defined parameter to avoid favoring either type of attribute. In order to overcome these problems, we present a novel clustering algorithm for mixed data, called DPC-MD. We improve DPC by using a new similarity criterion to deal with the three types of data: numerical, categorical, or mixed data. Compared to other methods for mixed data, DPC absolutely has more advantages to deal with non-spherical distribution data. In addition, the core of the proposed method is based on a new similarity measure for mixed data. This similarity measure is proposed to avoid feature transformation and parameter adjustment. The performance of our method is demonstrated by experiments on some real-world datasets in comparison with that of traditional clustering algorithms, such as K-Modes, K-Prototypes EKP and SBAC.

1. Introduction

Clustering analysis has attracted a lot of research attention due to its usefulness in many applications, including community detection, image processing, document processing, and so forth [1-7]. Clustering analysis has attracted a lot of research attention due to its usefulness in many applications, including most clustering algorithms rely on the assumption that data simply contain numerical values, but what should be dealt with is categorical values or mixed data containing both numerical and categorical values on data sets in the real world. For clustering algorithms dealing with mixed data, the core of these methods is how to measure the similarity for categorical attributes. Roughly, the existing clustering algorithms for mixed data can fall into two categories according to dealing with categorical attribute values. The first category of the methods is based on the pre-processing methods. The original attributes are transformed to new forms. Then, traditional distance functions are used to measure the transformed data in the new relation. The second category of approaches is based on similarity metrics dealing with categorical values directly.

Traditional clustering algorithms for mixed data have a pre-processing that is able to convert categorical attributes to new forms and facilitates processing. Binary encoding is the most common pre-processing method. This method transforms each categorical attribute to a set of binary attributes. For example, Ralambondrainy's algorithm [8] transforms categorical attributes into a set of binary attributes. Then, new forms are treated as numeric in the K-Means algorithm. Hence, we can directly adopt most traditional distances which are often used in numerical clustering, such as Euclidean distance, to define similarity between transformed objects. However, this method destructs the original structure of categorical attributes. In other words, transformed binary attributes are meaningless and their values are hard to interpret [9]. Apart from binary encoding, there are also other pre-processing methods. For example, in order to handle categorical data, Hsu [10] presents a new mechanism, distance hierarchy, which encodes a data set into a weighted tree structure. But it has a serious drawback that both the assignment of weights and the construction of distance hierarchies rely on domain knowledge.

In the respect of similarity metrics for categorical values, the K-Prototypes algorithm [11] is one of the most famous clustering algorithms for mixed data. Nevertheless, the choice of the weight γ has a significant effect on clustering results. As a variation of K-Prototypes algorithm, evolutionary K-Prototypes algorithm (EKP) [12], an unsupervised evolutionary clustering algo-

* Corresponding author at: School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, 221116, China.
E-mail address: dingsf@cumt.edu.cn (S. Ding).

http://dx.doi.org/10.1016/j.patrec.2017.07.001
0167-8655/

rithm for mixed type data, which integrates evolutionary computation framework with KP, also has a weight γ. Also, these algorithms [13,14] take into account the significance of different attributes towards the clustering process. However, a new parameter, the degree of fuzziness α, is introduced into these clustering algorithms. Hence, it will come out that choosing the parameter is a delicate and difficult task for users that may be a roadblock for using K-Prototypes and its variations efficiently. In addition, some algorithms [15,16] use entropy-type measures to group objects. However, these methods only deal with categorical data instead of mixed data and these entropy-type criteria can only measure the similarity between an object and a cluster. Besides, OCIL [17] gives a unified similarity metric which can be applied to mixed data using the entropy-based criterion. This similarity metric is also based on the concept of object-cluster similarity. In other words, it only can measure the similarity between an object and a cluster. In addition, OCIL is an iterative clustering algorithm. This means that this method requires a random initialization and may trap into local optimum. Similar to OCIL, Lim, et al. [18] propose a clustering framework for mixed attribute type dataset based on the entropy concept. It also needs to adjust the parameter which is used to balance attribute type between categorical attribute and numerical one. Besides, Li and Biswas [9] propose a Similarity-Based Agglomerative Clustering (SBAC) algorithm based on a new similarity metric that deals with the mixed data. But this method is high computational complexity and only suitable for some small data sets.

From the above discussion, most of clustering algorithms use the K-Means paradigm to cluster data having values. It means that those methods have an iterative process and probably trap into local optimum. A new algorithm, density peaks clustering (DPC) [19], proposed by Rodriguez and Laio is published in the US journal Science. This algorithm is able to detect non-spherical clusters without specifying the number of clusters. And more important, DPC does not need to iterate. Some studies [20–24] have been going on around this method. However, there are still some shortcomings. For example, DPC algorithm cannot find the correct number of clusters automatically. In order to overcome this difficulty, Liang and Chen [25] propose the 3DC clustering based on the divide-and-conquer strategy and the density-reachable concept. Du et al. [26] propose a density peaks clustering based on k nearest neighbors (DPC-KNN) which introduces the idea of k nearest neighbors (KNN) into DPC and has another option for the local density computation.

This paper presents a novel clustering algorithm, DPC-MD, based on a new similarity measure for mixed data. Actually, the proposed algorithm is the generalization of the original DPC algorithm. In order to assess the performance of the proposed algorithm, we compare the proposed algorithm with other algorithms on some UCI data sets. As a result, our algorithms have achieved satisfactory results in most data sets.

2. Related works

2.1. Notations

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ denote a dataset of n mixed data objects, where for each i, $1 \le i \le n$, $\mathbf{x}_i$ with m features consists of m_r numerical features and m_o categorical features. Therefore, for each i, $1 \le i \le n$, and for k, $1 \le k \le m_r$, let $x_{i,k}^{(r)}$ be the kth feature of $\mathbf{x}_i^{(r)}$, where $\mathbf{x}_i^{(r)}$ is the numerical part. Similarly, for each i, and for k, $1 \le k \le m_o$, $x_{i,k}^{(o)}$ denotes the kth feature of $\mathbf{x}_i^{(o)}$, where $\mathbf{x}_i^{(o)}$ is the categorical part. The domain of numerical feature $F_k^{(r)}$ is represented by continuous values. And categorical feature $F_k^{(o)}$ has t_k categories, i.e., $\mathrm{DOM}(F_k^{(o)}) = \{f_{k,1}, f_{k,2}, \cdots, f_{k,t_k}\}$, where $\mathrm{DOM}(F_k^{(o)})$ contains all possible values that can be chosen by attribute $F_k^{(o)}$. Therefore, $\mathbf{x}_i$ can be represented as $[\mathbf{x}_i^{(r)}, \mathbf{x}_i^{(o)}] = [x_{i,1}^{(r)}, x_{i,2}^{(r)}, \cdots, x_{i,m_r}^{(r)}, x_{i,m_r+1}^{(o)}, \cdots, x_{i,m}^{(o)}]$.

Distance functions such as Euclidean distance are used as similarity measure for numerical attribute. The Euclidean distance $\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j)$ between the object $\mathbf{x}_i$ and the object $\mathbf{x}_j$ is defined as:

$$\mathrm{dist}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2. \tag{1}$$

The definition of the information entropy $H(x)$ is given, as follow:

$$H(x) = -\sum_{x \in V} p(x) log(p(x)). \tag{2}$$

where $p(x)$ is the probability mass function of the random variable x. V is the finite set of possible outcomes of x.

2.2. Density peaks clustering

Its idea is that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. This method utilizes two important quantities: One is the local density ρ_i of each point $\mathbf{x}_i$, and the other is its distance δ_i from points of higher density. The two quantities correspond to two assumptions with respect to the cluster centers. One is that the cluster centers are surrounded by neighbors with a lower local density. The other is that they have relatively larger distance to the points of higher density. In the following, we will describe the computation of ρ_i and δ_i in much more detail.

DPC represents data objects as points in a space and adopts a distance metric, such as (1), as a similarity between objects.

The local density of a point $\mathbf{x}_i$, denoted by ρ_i, is defined as

$$\rho_i = \sum_j exp\left(-\frac{\mathrm{dist}\left(\mathbf{x}_i, \mathbf{x}_j\right)^2}{{d_c}^2}\right), \tag{3}$$

where d_c is an adjustable parameter, controlling the weight degradation rate.

d_c is the only variable in (3). The choice of d_c is actually the choice of the average number of neighbors of all points in data set. Let $v = n_d \times (p/100)$, where $n_d = \binom{n}{2}$ and p is a percentage. And n denotes the number of points in data set.

In the code presented by Rodriguez and Laio, d_c is define as

$$d_c = d_{\lceil \tau \rceil}, \tag{4}$$

where $d_{\lceil \tau \rceil} \in D = [d_1, d_1, \cdots d_{n_d}]$. D is a set of all the distances between every two points in data set, which are sorted in ascending order. $\lceil \tau \rceil$ is the subscript of $d_{\lceil \tau \rceil}$, where $\lceil \cdot \rceil$ is the ceiling function.

The computation of δ_i is quite simple. The minimum distance between the point of $\mathbf{x}_i$ and any other points with higher density, denoted by δ_i,

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i < \rho_j} \left(\mathrm{dist}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), if\ \exists j\ s.t. \rho_i < \rho_j \\ \max\limits_{j} \left(\mathrm{dist}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), otherwise \end{cases} \tag{5}$$

When the local density and delta values for each point have been calculated, this method identifies the cluster centers by searching anomalously large parameters ρ_i and δ_i. On the basis of this idea, cluster centers always appear on the upper-right corner of the decision graph.

After cluster centers have been found, DPC assigns each remaining points to the same cluster as its nearest neighbors with higher density. A representation named as decision graph is introduced to help one to make a decision. This representation is the plot of δ_i as a function of ρ_i for each point.

3. The proposed algorithm

We present a new similarity measure as a framework for handling mixed data with numerical and categorical attributes. To avoid generating an iterative process, we introduce the similarity metric to the density peaks clustering algorithm for clustering data.

3.1. Similarity measure

3.1.1. Similarity measure for numerical values

$\mathbf{x}_i^{(r)}$ and $\mathbf{x}_j^{(r)}$ are two objects with m_r numerical attributes. The similarity metric on numerical values can be calculated according to the Euclidean distance, i.e., formula (1). For ease of computing the similarity for mixed data, a generalization function, a monotonically decreasing function, is used to convert the distance *dist* into the judged similarity S_r [27,28]. In order to satisfy the condition: $dist(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)}) \in \Re \to S_r(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)}) \in [0, 1]$, we use a generalization function. Shepard [29] proposes a universal law of generalization for psychological science. It is applied to spatial generalization. Shepard's formulation is rather common. And it is given by an exponential function:

$$S_r\left(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)}\right) = \exp\left(-\text{dist}\left(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)}\right)^2/2\right). \tag{6}$$

The closer the value of S_r is to 1, the more similar the two objects are.

3.1.2. Similarity measure for categorical values

We define the similarity index between two objects $\mathbf{x}_i^{(o)}$ and $\mathbf{x}_j^{(o)}$ in terms of the categorical attribute $F_k^{(o)}$ as

$$\sigma\left(x_{i,k}^{(o)}, x_{j,k}^{(o)}\right) = \begin{cases} 1, if\ x_{i,k}^{(o)} = x_{j,k}^{(o)} \\ 0, if\ x_{i,k}^{(o)} \neq x_{j,k}^{(o)} \end{cases}. \tag{7}$$

We hope that the similarity for categorical part will range from 0 to 1. Again, we take account of the significance of each categorical attribute $F_k^{(o)}$. We have

$$S_o\left(\mathbf{x}_i^{(o)}, \mathbf{x}_j^{(o)}\right) = \sum_{k=1}^{m_o} w_k \sigma\left(x_{i,k}^{(o)}, x_{j,k}^{(o)}\right), \tag{8}$$

where $0 \le w_k \le 1$ and $\sum_{k=1}^{m_o} w_k = 1$. Obviously, w_k is the weight of categorical attribute $F_k^{(o)}$. In other words, w_k is the importance of categorical attribute $F_k^{(o)}$ contributing to the calculation of the similarity on the categorical attributes.

Now we discuss how to calculate the weight w_k of each categorical attribute $F_k^{(o)}$. We apply the notion of entropy to the calculation of the weights. We know that the inhomogeneity of the data set with respect to a categorical attribute corresponds to the importance of this categorical attribute from information theory. On the basis of Measure III in the paper [30], if the information content of an attribute is high, the inhomogeneity of the data set also is high for the attribute. The inhomogeneity of an attribute may be represented by the entropy of this attribute. As a result, for an attribute, more information content means more significance. In part, the entropy of each attribute reflects the weight w_k of the corresponding attribute. Therefore, according to (2), we can calculate the entropy of a categorical attribute $F_k^{(o)}$ with $\text{DOM}(F_k^{(o)}) = \{f_{k,1}, f_{k,2}, \cdots, f_{k,t_k}\}$ by

$$H_{F_k^{(o)}} = -\sum_{f_{k,l} \in DOM\left(F_k^{(o)}\right)} p\left(f_{k,l}\right) log\left(p\left(f_{k,l}\right)\right). \tag{9}$$

where the probability $p(f_{k,l})$ of attribute value $f_{k,l}$ can be calculated by $\sum_{i=1}^{n} \sigma(x_{i,k}^{(o)}, f_{k,l})/n$. Obviously, the numerator denotes the number of objects whose value of the categorical attribute $F_k^{(o)}$ equals to $f_{k,l}$. And, n is the total number of objects in the data set. Observing formula (9) carefully, we notice the fact that if the number of values chosen by $F_k^{(o)}$, t_k, is very large, then the entropy of this categorical attribute, $H_{F_k^{(o)}}$, is also high. This is not the same as the actual case. In order to lower the impact of the categorical attributes with too many different values or even unique values, such as the ID number, we redefine the entropy of a categorical attribute $F_k^{(o)}$ as

$$H'_{F_k^{(o)}} = -\frac{1}{t_k}\sum_{l=1}^{t_k} p\left(f_{k,l}\right) log\left(p\left(f_{k,l}\right)\right). \tag{10}$$

Hence, we can quantify the importance of a categorical attribute $F_k^{(o)}$ as

$$w_k = \frac{H'_{F_k^{(o)}}}{\sum_{k=1}^{m_o} H'_{F_k^{(o)}}}. \tag{11}$$

Substituting (11) into formula (8), we obtain the final similarity measure on the categorical attributes as follows:

$$S_o\left(\mathbf{x}_i^{(o)}, \mathbf{x}_j^{(o)}\right) = \sum_{k=1}^{m_o}\left(\frac{H'_{F_k^{(o)}}}{\sum_{k=1}^{m_o} H'_{F_k^{(o)}}} \cdot \sigma\left(x_{i,k}^{(o)}, x_{j,k}^{(o)}\right)\right). \tag{12}$$

Notice that, similar to $S_r(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)})$, the value of $S_o(\mathbf{x}_i^{(o)}, \mathbf{x}_j^{(o)})$ also falls into the interval [0, 1].

3.1.3. Similarity measure for mixed values

From the above content, it is easy to discover that we treat the similarity on the numerical part as whole, but calculate the similarity on the categorical part individually. Hence, this similarity between two mixed-type objects $\mathbf{x}_i$ and $\mathbf{x}_j$, denoted as $S(\mathbf{x}_i, \mathbf{x}_j)$, is defined by

$$\begin{aligned} S\left(\mathbf{x}_i, \mathbf{x}_j\right) = {} & \frac{m_r}{m}\exp\left(-\text{dist}\left(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)}\right)^2/2\right) \\ & + \frac{m_o}{m}\sum_{k=1}^{m_o}\left(\frac{H'_{F_k^{(o)}}}{\sum_{k=1}^{m_o} H'_{F_k^{(o)}}} \cdot \sigma\left(x_{i,k}^{(o)}, x_{j,k}^{(o)}\right)\right), \end{aligned} \tag{13}$$

where $m_r + m_o = m$ and the first term is the weighted similarity measure on the numerical attributes and the second term is the weighted similarity measure on the categorical attributes. Because the ranges of these two similarities $S_r(\mathbf{x}_i^{(r)}, \mathbf{x}_j^{(r)})$ and $S_o(\mathbf{x}_i^{(o)}, \mathbf{x}_j^{(o)})$ are the interval from 0 to 1, the value of $S(\mathbf{x}_i, \mathbf{x}_j)$ using the above weighting scheme also falls into the interval [0, 1].

To satisfy the requirement of the computation of the DPC algorithm, we convert the judged similarity $S(\cdot, \cdot)$ back into the distance $dist_u(\cdot, \cdot)$. The smaller the distance is, the more similar the two objects are. Hence, the distance measure finally can be defined as

$$\text{dist}_u\left(\mathbf{x}_i, \mathbf{x}_j\right) = -\log\left(S\left(\mathbf{x}_i, \mathbf{x}_j\right)\right). \tag{14}$$

3.2. The description of the algorithm

In this sub-section, we introduce the similarity metric presented in Section 3.1 to the DPC algorithm for handling mixed data. We can calculate the distance matrix for mixed data by the proposed similarity measure.

The following algorithm is a summary of the proposed DPC-MD.

3.3. Performance analysis

This sub-section analyzes the time complexity of the DPC-MD algorithm. Our proposed algorithm is the same as the DPC algo-

Algorithm 1 DPC-MD algorithm.

Inputs:
The samples $\boldsymbol{X} \in \Re_{n \times m}$
The parameter d_c
Outputs:
The label vector of cluster index: $\boldsymbol{Y} \in \Re_{n \times 1}$
Step 1. Calculate distance matrix according to (14)
Step 2. Calculate ρ_i for point $\mathbf{x}_i$ according to (3)
Step 3. Calculate δ_i for point $\mathbf{x}_i$ according to (5)
Step 4. Plot decision graph and select cluster centers
Step 5. Assign each remaining point to the cluster, which has its nearest neighbor of higher local density
Step 6. Return y

rithm and the only difference is the similarity measure. To be consistent with the above notations, we assume that n is the number of objects in the data set; m_r is the number of numerical attributes; m_o is the number of categorical attributes; t is the average number of different categorical attribute values. The computation cost of Step 1 is $O((m_r n)^2 + (t m_o n)^2)$. The direct implementation of Step 2 takes $O(n^2)$ time complexity. The implementation of the sorting process is $O(n \log n)$. As the complexity in assignment procedure is $O(n)$, the total time cost of this algorithm is $O((m_r n)^2 + (t m_o n)^2) + O(n^2) + O(n \log n) \sim O((t^2 m_o^2 + m_r^2) n^2)$.

4. Experiments and results

In this section, we use experimental results to exhibit the clustering performance and the robustness of our algorithm. In order to show the clustering performance of DPC-MD, we use it in some benchmark data sets with various mixed-type and categorical data sets. Almost all of the data sets are obtained from the UCI repository. On the categorical data sets, we compare the proposed algorithm with K-Modes [34], Evolutionary K-Prototypes algorithm (EKP) [12] and (Similarity-Based Agglomerative Clustering) SBAC [9] in accuracy. On the mixed-type data sets, we compare the proposed algorithm with K-Prototypes [11], EKP and SBAC in accuracy. Since the authors do not provide the implementation of this algorithm, we reimplement SBAC algorithm according to the paper [9]. It should be noted that we will treat the actual number of classes as prior information to facilitate the evaluation of the clustering results of this method. In other words, the number of clusters is given as the actual number of classes instead of the selection scheme used in the original paper.

We conduct experiments in a work station with a core i7 DMI2-Intel 3.6 GHz processor and 18GB RAM running MATLAB 2012B. In DPC and DPC-MD, we select the parameter d_c from [0.1% 0.2% 0.5% 1% 2% 4% 6%]. The parameter γ of the K-Prototypes and EKP varies from 0.1 to 2.1 in 0.1 increments. Due to using the random initialization, K-Prototypes, K-Modes and EKP are repeated 10 times.

4.1. Evaluation method

This paper uses clustering accuracy (ACC) [31–33] to measure the quality of clustering results. For n distinct samples $\mathbf{x}_i \in \Re^m$, y_i and c_i are the inherent category label and the predicted cluster label of $\mathbf{x}_i$, the calculation formula of ACC is

$$\mathrm{ACC} = \sum_{i=1}^{n} \sigma(\mathrm{y}_i, map(\mathrm{c}_i))/n, \tag{16}$$

where $map(\cdot)$ maps each cluster label to a category label by the Hungarian algorithm and this mapping is optimal. Like formula (7), $\delta(\mathrm{y}_i, map(\mathrm{c}_i))$ equal to 1 if $\mathrm{y}_i = map(\mathrm{c}_i)$ or 0 otherwise. In addition, N is the number of objects in the data set. The higher the ACC value is, the better the clustering performs.

Table 1
The details of the mixed data sets.

Data Sets	Cluster	Dimension ($m_r + m_o$)	Number
Credit Approval	2	6 + 9	653
Heart Disease	2	6 + 7	303
Australian Credit Approval	2	6 + 8	690
Lymphography	4	3 + 15	148
KDD Cup 1999	4	26 + 15	2000

Table 2
The details of the categorical data sets.

Data Sets	Cluster	Dimension	Number
Soybean	4	35	47
Congressional Voting	2	16	232
LED Display Domain	10	7	500

Table 3
Clustering accuracy of the evaluated algorithms on Credit Approval data set.

Algorithm	Clustering accuracy (ACC)	Parameter
DPC-MD	0.8407	$d_c = 6\%$
K-Prototypes	0.7796 ± 0.0390	$\gamma = 0.7, k = 2$
EKP	0.5513 ± 0	$\gamma = 1.3, k = 2$
SBAC	0.7525	$k = 2$

4.2. Experiments

The mixed-type data sets used in the experiment are all taken from the UCI Machine Learning Repository, including Credit Approval, Heart Disease, Australian Credit Approval and Lymphography. The details of these data sets are listed in Table 1.

The Categorical datasets used in the experiment are also taken from the UCI Machine Learning Repository, including Soybean, Congressional Voting Records and LED Display Domain. The details of these data sets are listed in Table 2.

4.2.1. Experiments on mixed datasets

There are a few missing values in the Credit Approval data set. A complete version of this data set has 690 objects. To facilitate handling this data set, we use a cleaned version (where objects with missing values are not included) with 653 objects. Thus, the Credit Approval data set consists of 653 samples with six numerical and nine categorical attributes. The data objects can be divided into two classes. In Table 3, we list the clustering accuracy of our proposed algorithm, K-Prototypes, EKP and SBAC on this set. In this case, the clustering accuracy values of K-Prototypes, EKP and SBAC are 0.7796, 0.5513, 0.7525, respectively, as shown in Table 3. The ACC value of our algorithm is 0.8407 at $d_c = 6\%$. Experimental results on the Credit Approval data set show that the ACC value of DPC-MD is 6.11%, 28.94%, 8.82% higher than K-Prototypes, EKP and SBAC respectively.

Table 4
Clustering accuracy of the evaluated algorithms on Heart Disease data set.

Algorithm	Clustering accuracy (ACC)	Parameter
DPC-MD	0.8218	$d_c = 0.1\%$
K-Prototypes	0.7812 ± 0.0386	$\gamma = 0.4, k = 2$
EKP	0.5743 ± 0.0104	$\gamma = 0.2, k = 2$
SBAC	0.7525	$k = 2$

Table 5
Clustering accuracy of the evaluated algorithms on Australian Credit Approval data set.

Algorithm	Clustering accuracy (ACC)	Parameter
DPC-MD	0.8652	$d_c = 6\%$
K-Prototypes	0.7925 ± 0.0295	$\gamma = 0.4, k = 2$
EKP	0.5590 ± 0.0014	$\gamma = 1.1, k = 2$
SBAC	0.6000	$k = 2$

Table 6
Clustering accuracy of the evaluated algorithms on the Lymphography data set.

Algorithm	Clustering accuracy (ACC)	Parameter
DPC-MD	0.6149	$d_c = 0.1\%$
K-Prototypes	0.4818 ± 0.0588	$\gamma = 1.5, k = 4$
EKP	0.5797 ± 0.0634	$\gamma = 0.2, k = 4$
SBAC	0.5676	$k = 4$

Table 7
Clustering accuracy of the evaluated algorithms on the KDD Cup 1999 data set.

Algorithm	Clustering accuracy (ACC)	Parameter
DPC-MD	1.0	$d_c = 6\%$
K-Prototypes	0.7500 ± 0	$\gamma = 1.5, k = 4$
EKP	0.4805 ± 0.0789	$\gamma = 2, k = 4$
SBAC	–	–

The Heart Disease data set consists of 303 samples with six numerical and seven categorical attributes. The data objects can be divided into two classes. Table 4 shows that the ACC value of DPC-MD is 4.06%, 24.75%, 6.93% higher than K-Prototypes, EKP and SBAC respectively. Note that EKP produces the stable results on the data set. The results verifies EKP's advantage that it is not sensitive to initialization.

The Australian Credit Approval data set consists of 690 samples with six numerical and eight categorical attributes. The data objects can be divided into two classes. Table 5 shows that the ACC value of DPC-MD is 7.27%, 30.62%, 26.52% higher than K-Prototypes, EKP and SBAC respectively.

The Lymphography data set consists of 148 samples with three numerical and fifteen categorical attributes. The data objects can be divided into four classes. Table 6 shows that the ACC value of DPC-MD is 13.31%, 3.52%, 4.73% higher than K-Prototypes, EKP and SBAC respectively.

A complete version of the KDD Cup 1999 data set has 4,000,000 objects. Our device cannot process such a large data set due to the limitations of the memory. Since we use a subset of the KDD Cup 1999 data set has 2000 objects, equally distributed into four classes. And each object still has 26 numerical and 15 categorical attributes Table 7 shows that the ACC value of DPC-MD is 25%, 51.92% higher than K-Prototypes and EKP respectively. The symbol - means that we do not get the result due to both high computational complexity of SBAC and the limitations of our device. Note that SBAC spends a lot of time finding the uncommon feature, especially when dealing with large data sets with high-dimensional numerical attributes.

Table 8
Clustering accuracy of the evaluated algorithms on the Soybean data set.

Algorithm	Clustering accuracy (ACC)	Parameter
DPC-MD	1.0	$d_c = 6\%$
K-Modes	0.7787 ± 0.1683	$k = 4$
EKP	0.9596 ± 0.0067	$k = 4$
SBAC	0.2979	$k = 4$

Table 9
Clustering accuracy of the evaluated algorithms on the Congressional Voting Records data set.

Algorithm	Clustering accuracy (ACC)	Parameter
DPC-MD	0.9138	$d_c = 1\%$
K-Modes	0.8694 ± 0.0054	$k = 2$
EKP	0.8664 ± 0	$k = 2$
SBAC	0.5388	$k = 2$

Table 10
Clustering accuracy of the evaluated algorithms on the LED Display Domain data set.

Algorithm	Clustering accuracy (ACC)	Parameter
DPC-MD	0.6860	$d_c = 0.1\%$
K-Modes	0.5310 ± 0.0568	$k = 10$
EKP	0.6312 ± 0.0482	$k = 10$
SBAC	0.3380	$k = 10$

As can be seen from Tables 3 to 7, experimental results of DPC-MD are significantly better than those obtained by other methods for these data sets.

4.2.2. Experiments on categorical datasets

The Soybean data set consists of 47 samples with 35 categorical attributes. The data objects can be divided into four classes. Table 8 shows that the ACC value of DPC-MD is 22.13%, 4.04%, 70.21% higher than K-Modes, EKP and SBAC respectively.

Similar to the Credit Approval data set, there are a few missing values in the Congressional Voting Records data set. A complete version of this data set has 435 objects. In contrast, we use a cleaned version consisting of 232 objects. The Congressional Voting Records data set consists of 232 samples with 16 categorical attributes. The data objects can be divided into two classes. Table 9 shows that the ACC value of DPC-MD is 4.44%, 4.74%, 37.50% higher than K-Modes, EKP and SBAC respectively.

The LED Display Domain data set is a sample of 500 objects obtained from the original data generator. Thus, the LED Display Domain data set consists of 500 samples with 7 categorical attributes. The data objects can be divided into ten classes. Table 10 shows that the ACC value of DPC-MD is 15.50%, 5.48%, 34.80% higher than K-Modes, EKP and SBAC respectively.

As can be seen from Tables 8 to 10, K-Modes and EKP are conducted repeatedly, because DPC-MD and SBAC without initialization come out stable clustering results when the parameter is given. Obviously, the clustering results obtained by our algorithm are, in most of the cases, superior to the one obtained by the other methods.

In conclusion, on these categorical and mixed-type data sets, the ACC values obtained by DPC-MD are superior to those obtained by other methods. The main reason for this is that K-Modes, K-Prototypes and EKP are sensitive to initialization and are unsuitable for non-spherical distribution data. SBAC proposes the similarity measure based on the assumption that the more uncommon matched feature value corresponds to greater weight. We conjecture that this assumption is not appropriate for these data sets. Due to these factors, these comparison partners do not have ex-

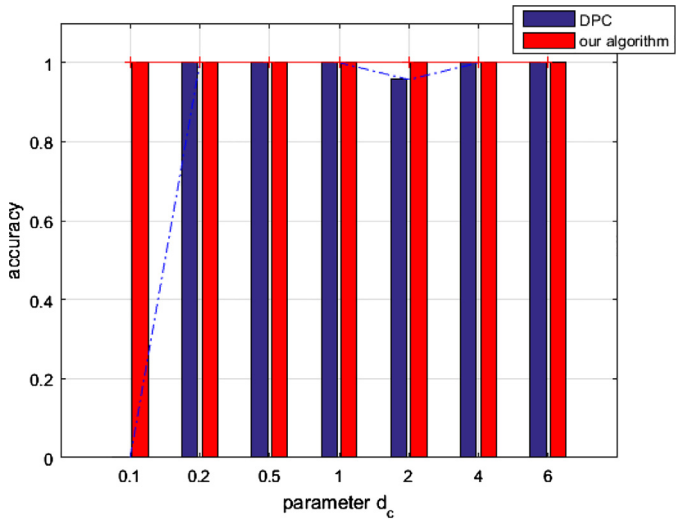


Fig. 1. Clustering accuracy of DPC and our algorithm on the Soybean data set.

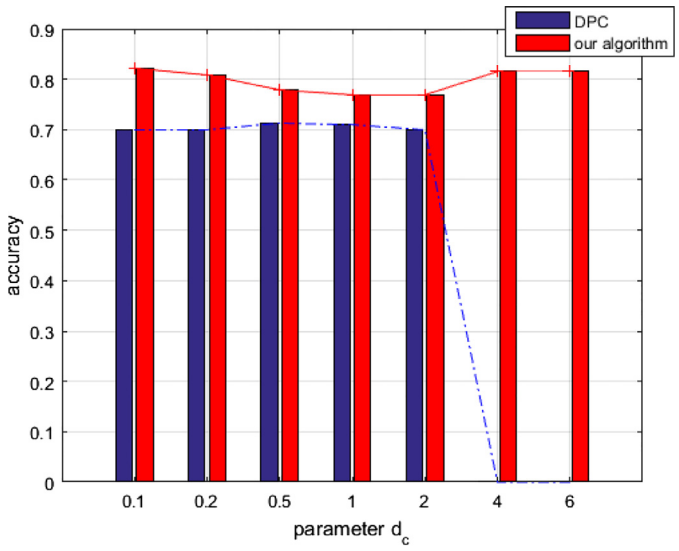


Fig. 3. Clustering accuracy of DPC and our algorithm on the Heart Disease data set.

cellent jobs. The proposed method overcomes above problems effectively. Thus it obtains good clustering results on these data sets. Even, DPC-MD finds the optimal structure of clusters in clustering the Soybean data set.

4.3. Robustness tests

As is known to all, the density peaks clustering algorithm is developed to cluster numerical data. In order to facilitate comparing our method with DPC, we use a pre-processing method which make DPC work on mixed data sets. More specifically, we convert categorical values to integer values on categorical data sets and mixed-type data sets. Thus we can directly adopt Euclidean distance to compute dissimilarity between transformed objects.

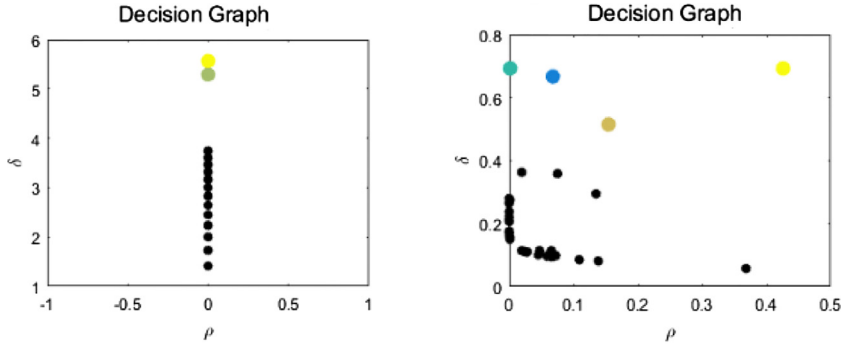
Due to space constraints, only two of the data sets be explored here. Fig. 1 shows the clustering accuracy of DPC and DPC-MD on the Soybean data set with varying parameter d_c. In this figure, we can see that DPC generates good results in most cases, even the worst value is 0.9574. Nevertheless, DPC cannot work when d_c equals to 0.1 (This reason is discussed in detail below). By contrast, the clustering results of DPC-MD are stable no matter what value the parameter d_c is.

In some cases, DPC does a poor job of finding the clusters, which we need to pay extra attention to. Fig. 2(a) shows that the decision graph is produced by DPC on the Soybean data set, when the parameter is 0.1%. Only three cluster centers can be found by DPC on decision graph. In this case, we are incapable of making the right choices. It means that DPC generates wrong number of clusters on this data set. In contrast, DPC-MD using the proposed similarity generates correct number of clusters, when the parameter is 0.1%. Fig. 2(b) shows the corresponding decision graph.

Fig. 3 shows the clustering accuracy DPC and DPC-MD on the Heart Disease data set with varying parameter d_c. DPC cannot work on this data set occasionally, too. In this figure, the curve of our algorithm is higher than the curve of DPC.

These figures show that the curves of our algorithm is almost flat, or even completely flat. The above experiments demonstrate that the choice of the parameter d_c has only a minor impact on the clustering results of our algorithm. In other words, the proposed algorithm has strong robustness.

Beyond the decision graph, Rodriguez and Laio present a γ-graph. It provides a hint for choosing the number of centers using the plot of $\gamma = \rho\delta$ sorted in decreasing order. Fig. 4 displays the γ-graph results of DPC and our algorithm for clustering the Heart Disease data set. Fig. 4(a) and (c) show the γ-graph results of DPC. It can be easily noticed that the blue dot (represents one of two centers) is far away from the “straight line” (represents other points), whereas it is hard to separate the yellow dot (represents the other center) from other points. Consequently, on the Heart Disease set, when the parameter is 4% or 6%, DPC hardly finds correct number of clusters by using the γ-graph. As shown in Fig. 4(b) and (d), the γ-graph results of our algorithm show that the global



(a) $DPC\ d_c = 0.1\%$ (b) $DPC-MD\ d_c = 0.1\%$

Fig. 2. The decision graphs on the Soybean set.

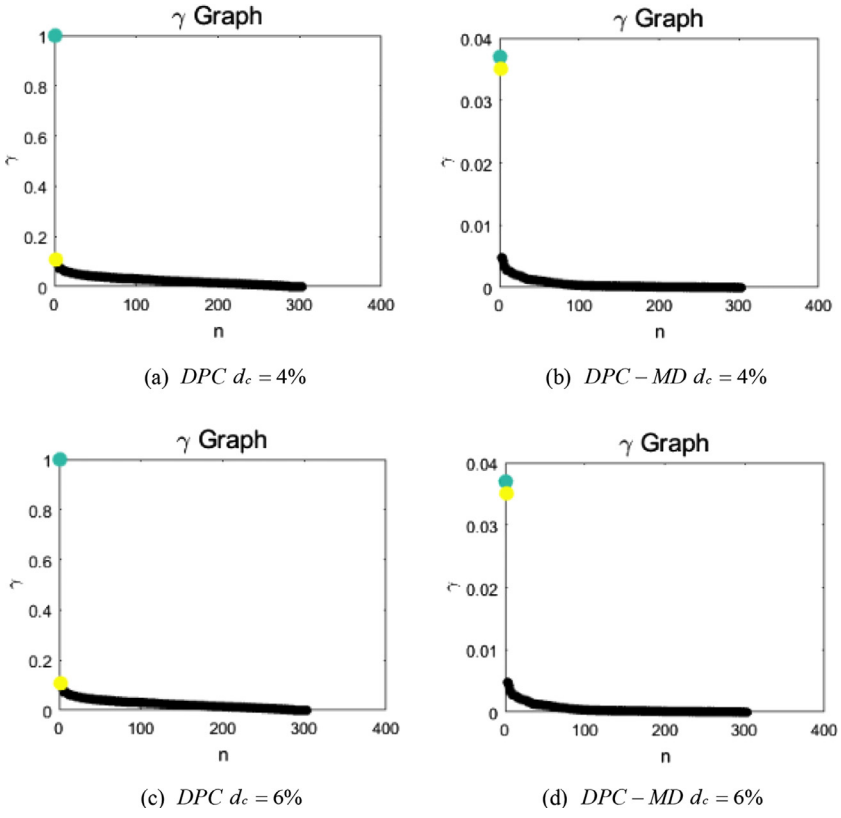


Fig. 4. The γ-graphs of DPC and DPC-MD on the Heart data set.

distribution of the points is a "straight line", but the centers of the two clusters are the outliers deviating from the global distribution.

The combination of our proposed algorithm and the γ-graph displays a possible that the proposed algorithm can automatically determine the cluster centers. However, this is not the research focus of this article. Hence, we will not further discuss this in this article.

Experimental results prove that DPC-MD obtains more robust performance than the original algorithm. The reason for this is that the pre-processing destructs the original structure of categorical attributes. The component of transformed categorical attributes is also measured by Euclidean distance, so the dissimilarity metric does not reveal the dissimilarity between categorical values. Especially, when data sets have categorical attributes with hundreds or thousands of categories, compared with two distinct values may yield a very large difference, while it can also yield a difference of zero. By contrast, our proposed similarity metric can reveal the structure of the clusters better.

5. Conclusions

We present a similarity metric for measuring numerical and categorical values. Robustness tests prove that the proposed similarity metric can better reveal the structure of the data sets than pre-processing methods. The similarity metric used in K-Prototypes and EKP needs to choose the parameter γ to avoid favoring either type of attribute. However, it comes out that choosing the parameter is a delicate and difficult task for users. In contrast, our similarity metric can circumvent parameter adjustment effectively. Although the similarity metric used in SBAC also does not need to adjust the parameter, a computational efficient similarity measure remains to be developed. In addition, experiment results show the assumption used in similarity computations does not apply to all the data sets. To better detect non-spherical distribution, we integrate the similarity metric with the density peaks clustering algorithm without initialization. We further bring forward a new clustering algorithm for handling mixed data, called DPC-MD. The experimental results support our claim that DPC-MD is an efficient algorithm for clustering mixed data. Besides, we improve the robustness of the original algorithm. This means that clustering results are less sensitive to the choice of the parameter d_c. More importantly, we find it possible that the proposed algorithm can automatically determine the cluster centers based on the γ-graph.

Future works will develop an automatic cluster centroid selection method on the basis of the proposed algorithm. Despite the proposed algorithm has great stability with different values of d_c, this method also needs to determine the value of the parameter. In future works, we will focus on investigating this problem that we can automatically determine the parameter d_c.

Acknowledgment

This work is supported by "the Fundamental Research Funds for the Central Universities" (No. 2017XKZD03).

References

[1] L. Chen, Z. Xu, H. Wang, S. Liu, An ordered clustering algorithm based on K-means and the PROMETHEE method, Int. J. Mach. Learn. Cybern. (2016), doi:10.1007/s13042-016-0617-9.
[2] J. Ma, D. Tian, M. Gong, L. Jiao, Fuzzy clustering with non-local information for image segmentation, Int. J. Mach. Learn. Cybern. 5 (6) (2014) 845–859.
[3] Y. Zheng, B. Jeon, D. Xu, Q.M. Wu, H. Zhang, Image segmentation by generalized hierarchical fuzzy C-means algorithm, J. Intell. Fuzzy Syst. 28 (2) (2015) 961–973.
[4] S. Zeng, X. Yang, J. Gou, J. Wen, Integrating absolute distances in collaborative representation for robust image classification, CAAI Trans. Intell. Technol. 1 (2) (2016) 189–196.
[5] Z. Xia, X. Wang, X. Sun, Q. Liu, N. Xiong, Steganalysis of LSB matching using differences between nonadjacent pixels, Multimedia Tools Appl. 75 (4) (2016) 1947–1962.

[6] L. Liu, Y. Guo, Z. Wang, Z. Yang, Y. Shao, k-Proximal plane clustering, Int. J. Mach. Learn. Cybern. (2016), doi:10.1007/s13042-016-0526-y.
[7] X. Wen, L. Shao, Y. Xue, W. Fang, A rapid learning algorithm for vehicle classification, Inf. Sci. 295 (2015) 395–406.
[8] H. Ralambondrainy, A conceptual version of the K-means algorithm, Pattern Recognit. Lett. 16 (11) (1995) 1147–1157.
[9] C. Li, G. Biswas, Unsupervised learning with mixed numeric and nominal data, IEEE Trans. Knowl. Data Eng. 14 (4) (2002) 673–690.
[10] C.C. Hsu, Generalizing self-organizing map for categorical data, IEEE Trans. Neural Netw. 17 (2) (2006) 294–304.
[11] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Min. Knowl. discovery 2 (3) (1998) 283–304.
[12] Z. Zheng, M. Gong, J. Ma, L. Jiao, Q. Wu, Unsupervised evolutionary clustering algorithm for mixed type data, in: Proceedings of 2010 IEEE Congress on Evolutionary Computation, 2010.
[13] S.P. Chatzis, A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional, Expert Syst. Appl. 38 (7) (2011) 8684–8689.
[14] J. Ji, W. Pang, C. Zhou, X. Han, Z. Wang, A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data, Knowl.-Based Syst. 30 (2012) 129–135.
[15] D. Barbará, Y. Li, J. Couto, COOLCAT: an entropy-based algorithm for categorical clustering, in: Proceedings of the Eleventh International Conference on Information and Knowledge Management, 2002, pp. 582–589.
[16] T. Li, S. Ma, M. Ogihara, Entropy-based criterion in categorical clustering, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, pp. 536–543.
[17] Y.M. Cheung, H. Jia, A unified metric for categorical and numerical attributes in data clustering, in: Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2013, pp. 135–146.
[18] J. Lim, J. Jun, S.H. Kim, D.A. McLeod, Framework for clustering mixed attribute type datasets, in: Proceedings of the 4th International Conference on Emerging Databases, 2012.
[19] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[20] G. Chen, X. Zhang, Z.J. Wang, F. Li, Robust support vector data description for outlier detection with noise or uncertain data, Knowl.-Based Syst. 90 (2015) 129–137.
[21] K. Lu, S. Xia, C. Xia, Clustering based road detection method, in: Proceedings of the 34th Chinese Control Conference, 2015, pp. 3874–3879.
[22] B. Wang, J. Zhang, Y. Liu, Y. Zou, Density peaks clustering based integrate framework for multi-document summarization, CAAI Trans. Intell. Technol. 2 (1) (2017) 26–30.
[23] M. Du, S. Ding, X. Xu, Y. Xue, Density peaks clustering using geodesic distances, Int. J. Mach. Learn. Cybern. (2017), doi:10.1007/s13042-017-0648-x.
[24] J. Xu, G. Wang, T. Li, W. Deng, G. Gou, Fat node leading tree for data stream clustering with density peaks, Knowl.-Based Syst. 120 (2017) 99–117.
[25] Z. Liang, P. Chen, Delta-density based clustering with a divide-and-conquer strategy: 3DC clustering, Pattern Recognit. Lett. 73 (2016) 52–59.
[26] M. Du, S. Ding, H. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, Knowl.-Based Syst. 99 (2016) 135–145.
[27] F.G. Ashby, N.A. Perrin, Toward a unified theory of similarity and recognition, Psychol. Rev. 95 (1) (1988) 124–150.
[28] S. Santini, R. Jain, Similarity measures, IEEE Trans. Pattern Anal. Mach. Intell. 21 (9) (1999) 871–883.
[29] R.N. Shepard, Toward a universal law of generalization for psychological science, Science 237 (4820) (1987) 1317–1323.
[30] J. Basak, R. Krishnapuram, Interpretable hierarchical clustering by constructing an unsupervised decision tree, IEEE Trans. Knowl. Data Eng. 17 (1) (2005) 121–132.
[31] Y. Zhang, X. Sun, B. Wang, Efficient algorithm for K-Barrier coverage based on integer linear programming, China Commun. 13 (7) (2016) 16–23.
[32] N. Iam-On, T. Boongoen, N. Kongkotchawan, A new link-based method to ensemble clustering and cancer microarray data analysis, Int. J. Collaborative Intell. 1 (1) (2014) 45–67.
[33] K.I. Qazi, H.K. Lam, B. Xiao, G. Ouyang, X. Yin, Classification of epilepsy using computational intelligence techniques, CAAI Trans. Intell. Technol. 1 (2) (2016) 137–149.
[34] Z. Huang, A fast clustering algorithm to cluster very large categorical data sets in data mining, in: Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997.