Applied Soft Computing xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

# A novel ant-based clustering algorithm using Renyi entropy

Q1 Lei Zhang[a,*], Qixin Cao[a], Jay Lee[b]

[a] Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai 200240, China
[b] NSF Center for Intelligent Maintenance Systems, University of Cincinnati, OH 45221,USA

ARTICLE INFO

*Article history:*
Received 27 May 2011
Received in revised form 31 October 2012
Accepted 22 November 2012
Available online xxx

*Keywords:*
Swarm intelligence
Ant-based clustering
Renyi entropy
Kernel
The Friedman test

ABSTRACT

Ant-based clustering is a type of clustering algorithm that imitates the behavior of ants. To improve the efficiency, increase the adaptability to non-Gaussian datasets and simplify the parameters of the algorithm, a novel ant-based clustering algorithm using Renyi Entropy (NAC-RE) is proposed. There are two aspects to application of Renyi entropy. Firstly, Kernel Entropy Component Analysis (KECA) is applied to modify the random projection of objects when the algorithm is run initially. This projection can create rough clusters and improve the algorithm's efficiency. Secondly, a novel ant movement model governed by Renyi entropy is proposed. The model takes each object as an ant. When the object (ant) moves to a new region, the Renyi entropy in its local neighborhood will be changed. The differential value of entropy governs whether the object should move or be moveless. The new model avoids complex parameters that have influence on the clustering results. The theoretical analysis has been conducted by kernel method to show that Renyi entropy metric is feasible and superior to distance metric. The novel algorithm was compared with other classic ones by several well-known benchmark datasets. The Friedman test with the corresponding Nemenyi test are applied to compare and conclude the algorithms' performance The results indicate that NAC-RE can get better results for non-linearly separable datasets while its parameters are simple.

## 1. Introduction

Swarm intelligence is one kind of intelligent behavior shown by the cooperation of collective insects, such as ants and bees. Since 1990, several collective behavior inspired algorithms have been proposed. Particle Swarm Optimization (PSO) [1,2] and Ant Colony Optimization (ACO) [3,4] are the most popular in this domain. Recently, prey models [5] also show an increase popularity. PSO is designed to simulate the choreography of bird flocking. The birds are represented by a population of particles and each particle has a certain location and velocity within the search space [1]. Particles fly through the search space in search of high quality solutions. ACO is inspired by the behavior of ant colonies for food searching. The pheromone trails between the ants enables them to find the shortest path between their nest and food source [3,4]. In the prey model, a forager needs to decide whether to attack the prey or to continue searching. The foraging agent should maximize the energy intake with respect to the probability to attack [5].

The application areas of these algorithms include NP hard optimization problems (such as the traveling salesman problem), the quadratic assignment, the network routing, clustering and job scheduling. The general review of swarm intelligence in data mining such as rule induction, classification and clustering, can refer to [6,7]. In this paper, we mainly focus the algorithms to imitate the ants' behavior for clustering purpose.

Clustering is a method that divides a dataset into groups of similar objects, thereby minimizing the similarities between different clusters and maximizing the similarities between objects in the same cluster. Clustering is widely applied in data mining, such as in document clustering and Web analysis. Classic clustering approaches include partition-based methods, such as K-means, K-medoids, and K-prototypes [8,9]; hierarchy-based methods, such as BIRCH [10]; density-based methods such as LDBSCAN [11,12]; grid-based methods such as GGCA [13]; and model-based methods, such as neural networks and Self-Organizing Map (SOM) [14,15].

Recently, ant-based clustering, which is a type of clustering algorithm that imitates the behavior of ants, has earned researchers' attention. Ant-based clustering can be divided into two classes. The first class imitates the ant's foraging behavior, which involves finding the shortest route between a food source and the nest. This intelligent behavior is achieved by means of pheromone trails and information exchange between ants [16,17]. The algorithms treat clustering as an optimization task and utilize ACO methods to obtain optimal clusters. A variant of ACO, called the Aggregation Pheromone density-based Clustering algorithm (APC), was also suggested [18]. Similar to ACO, APC is based on the aggregation pheromones found in ants. The advantage of these methods is that the objective function is explicit. The key elements of these

[*] Corresponding author.
*E-mail addresses:* zhanglei@sjtu.edu.cn, zhanglei75@sina.com (L. Zhang).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.11.022

algorithms are the pheromone matrix updating rule and the heuristic function.

The second class imitates ants' behavior of clustering their corpses and forming cemeteries. Some ants can pick up dead bodies randomly distributed in the nest and group them into different sizes. The large group of bodies attracts the ants to deposit more dead bodies and becomes larger and larger. The essence of this phenomenon is positive feedback [19]. One of the first studies related to this domain is the work of Deneubourg [20], who came up with the Basic Model (BM) to explain the ants' movement. In the BM, the ants move randomly and pick up or drop objects according to the number of similar surrounding objects to cluster them. Lumer and Faieta [21] extended the model and applied it to data analysis (they called this the LF algorithm). In their analysis, an object with $n$ attributes can be viewed as a point in the $R^n$ space. The point is projected into a low-dimensional space (often a two-dimensional plane). The similarity of the object with those in the local neighborhood is calculated to determine whether the object should be picked up or dropped by ants. As a basic algorithm, LF was followed and improved by a number of modified algorithms in different applications. Wu et al. [22] further explained the idea of the similarity coefficient (this coefficient defines the scale for objects' similarity) and suggested a more simple probability conversion function. Ramos and Merelo [23] studied ant-based clustering with different ant speeds to cluster text documents. Yang et al. [24,25] suggested multiple ant colonies consisting of independent colonies and a queen ant agent. Each ant colony had a different moving speed and probability conversion function. The hypergraph model was used to combine the results of all parallel ant colonies.

In addition to the above-mentioned studies, a series of research by Handl deserves special attention. She came up with a set of strategies for increasing the robustness of the LF algorithm and applying it to document retrieval [26]. She performed a comparative study of ant-based clustering with K-means, average links, and 1d-SOM [27,28]. An improved version, ATTA, which incorporates adaptive and heterogeneous ants and time-dependent transporting activity, was proposed in her latest paper [29]. The main feature of this kind of algorithm is that the algorithm directly imitates the ant's behavior to cluster data and the clustering objective is implicitly defined [30].

Beyond these two classes of ant-based clustering, Tsang and Kwong [31] proposed Ant Colony Clustering for anomaly intrusion detection. This method integrates the characteristics of the two above-mentioned classes. Specifically, cluster formation and searching for an object are regarded as nest building and food foraging, respectively. The ant exhibits picking up and dropping behaviors while simultaneously depositing cluster-pheromones on the grid. Xu et al. [32] suggested a novel ant movement model wherein each object was viewed as an ant. The ant determines its behavior according to the fitness of its local neighborhood. Essentially, this model is similar to that in the second class of ant-based clustering.

Combinations of ant-based clustering with other clustering methods can also be found. For example, ant-based clustering has been combined with K-means [33] and with K-harmonic means [34]; ant colonies have been hybridized with fuzzy C-means [35]; fuzzy ants have been endowed with intelligence in the form of IF-THEN rules [36]; and the hybrid approach has been generated based on Particle Swarm Optimization (PSO), ACO, and K-means [37]. In these methods, the role of ant-based clustering is mainly to create initial clusters for other clustering algorithms.

A comprehensive overview of ant-based and swarm-based clustering can be found in [30]. Our particular interest is in the second kind of ant-based clustering discussed above. The process of this kind of algorithms can be generalized as five steps (detailed description is in Section 2):

(1) *Projection*: All objects and ants are randomly projected onto the toroidal grid.
(2) *Calculating the similarity*: Each ant calculates the object's similarity to others in the object's local neighborhood.
(3) Picking up or dropping objects.
(4) Ants move.
(5) Repeat (2)–(4).

Although this kind of ant-based clustering has been modified gradually, there are still some problems needed to be solved. The focus of our work is on the following three important problems.

- Improving the algorithm's efficiency
  It is not highly efficient because of the randomness in the algorithm. Because the objects are randomly projected onto the toroidal grid at the initial time of the algorithm, the similarities of the objects in a local neighborhood are very low. Therefore, the objects are easily picked up but not easily dropped by the ants. It takes a long time to go from the inception of the algorithm to the moment when the rough clusters are created. Commonly, tens of thousands of iterations are needed for ant-based clustering algorithms [17,29,39].
- Improving the adaptability of the algorithm to the datasets with special structures
  In the essence, ant-based clustering algorithms are distance-based because the similarity of the objects is computed by Euclidean distance or Cosine distance. Just like other distance-based clustering algorithms, it is effective for the datasets with ellipsoidal or Gaussian structure. If the separation boundaries between clusters are nonlinear, it will fail [38].
- Simplifying the parameters in the algorithm
  There are several parameters in ant-based clustering, such as the similarity coefficient, the constants in the probability conversion functions (which will be described in Section 2). Some parameters are difficult to set properly, while they have an important effect on the clustering results. For example, a too small choice of the similarity coefficient $\alpha$ prevents the formation of clusters; on the other hand, a too large choice results in the fusion of individual clusters [22,26–29]. As mentioned in [39], the complex parameter setting should be avoided to simplify the use of the algorithm.

To solve these problems, a novel ant-based clustering algorithm integrated with Renyi entropy (NAC-RE) is proposed. The applications of Renyi entropy in NAC-RE are shown in two aspects. First, Kernel Entropy Component Analysis (KECA) is used to modify the initial projection of all objects. Second, a novel ant movement model governed by Renyi entropy is created. These two applications are geared toward solving the problems mentioned above.

Various attempts have been made to utilize information theory in clustering [40,41]. Tsang and Kwong [31] first introduced the application of local regional entropy in ant-based clustering. Liu et al. [38] proposed entropy-based metrics in ant-based clustering. Entropy governs the ant's picking up and dropping behaviors. They pointed out that entropy-based ant clustering required fewer training parameters than density-based. However, they used traditional Shannon entropy. First, the attributes of the objects must be independent. Second, the computation of the entropy needs discretization of each attribute of the object. They did not indicate how to set the resolution of each attribute. Different from their work, we use Renyi entropy in our study. Renyi entropy lends itself nicely to non-parametric estimation and overcomes the difficulty in computing Shannon entropy [42]. In our proposed method, each object

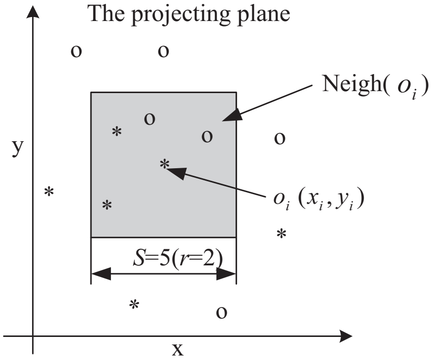


**Fig. 1.** the local neighborhood of the object $o_i$.

is taken as an ant, and the ant's movement was governed by the change of Renyi entropy in its local neighborhood. Because Renyi entropy is only dependent on the dataset, the parameters in the algorithm are simple. Meanwhile, entropy metric takes the place of distance metric, which makes the algorithm superior in clustering some nonlinearly separable datasets.

The remainder of this paper is organized as follows: Section 2 describes the basics and problems of ant-based clustering algorithm. Section 3 introduces Renyi entropy, KECA and their applications in clustering. Section 4 proposes a novel ant-based clustering algorithm using Renyi entropy (NAC-RE). Section 5 provides the theoretical analysis of NAC-RE algorithm using kernel method. Section 6 reports the evaluating results of NAC-RE algorithm compared with other algorithms. Finally, Section 7 gives the conclusions and future work.

## 2. The basics and problems of ant-based clustering algorithm

The algorithm introduced by Lumer and Faieta [21] represents the basic ant-based clustering method. Some important concepts are firstly introduced through Fig. 1.

*The projecting plane*: The objects and ants are initially projected onto a two-dimensional plane. Each object or ant is projected randomly. The size of the plane can be determined based on the number of objects.

The local neighborhood of object $o_i$: It is a neighboring region of the object $o_i$ and written as *Neigh*($o_i$). It is often a square with size $s \times s (s = 2r + 1)$, where $r$ is the radius of *Neigh*($o_i$). The center of *Neigh*($o_i$) is the position of $o_i$.

*The local similarity*: The similarity of the object $o_i$ with other objects in *Neigh*($o_i$). It is often measured by the distance between objects. In Fig. 1, assume that an ant finds an object $o_i$ at the coordinates ($x_i$,$y_i$). The local similarity of $o_i$ is given by

$$f(o_i) = \begin{cases} \dfrac{1}{s^2} \displaystyle\sum_{o_j \in Neigh(o_i)} \left[ 1 - \dfrac{d(o_i, o_j)}{\alpha} \right], \text{when} f > 0 \\ 0 \,\text{otherwise} \end{cases}, \qquad (1)$$

where $d(o_i,o_j)$ is the distance between two objects. Typically Euclidean distance is used. $\alpha$ is a factor that defines the scale for dissimilarity. The setting of $\alpha$ is an important research focus in the publications [22,26,27,29,39].

*The probability conversion function*: It is a function that converts the local similarity of $o_i$ into the probability of being picked up (or dropped) by ants. The probability that an ant will pick up or drop the object is

$$P_p(o_i) = \left( \frac{k_1}{k_1 + f(o_i)} \right)^2, \qquad (2)$$

**Table 1**
The pseudo-code of the LF algorithm.

```
/*initialization*/
Randomly scatter the objects on the toroidal grid
Randomly place the ants on the toroidal grid
Initialize all parameters: $r$,$t_{\max}$,$\alpha$,$k_1$,$k_2$
/* main loop*/
for t = 1 to $t_{\max}$ do
    for all ants do
        if (ant unladed) and (grid occupied by object $o_i$) then
            compute $f(o_i)$ and $P_p(o_i)$
            draw a random real number $p \in (0, 1)$
            if ($p \le P_p(o_i)$) then
                pick up object $o_i$
            end if
        else
            if (ant carrying object $o_i$) and (grid empty) then
                compute $f(o_i)$ and $P_d(o_i)$
                draw a random real number $p \in (0, 1)$
                if ($p \le P_d(o_i)$) then
                    drop object $o_i$
                end if
            end if
        end if
        Randomly move to one neighboring grid not occupied by other ants
    end for
    adjust $r$, $\alpha$
end for
Output locations of all objects;
```

$$P_d(o_i) = \begin{cases} 2f(o_i) \,\text{when} f(o_i) < k_2 \\ 1 \quad \text{when} f(o_i) \ge k_2 \end{cases}, \qquad (3)$$

where $k_1, k_2$ are two constants. $k_1$ and $k_2$ adjust the probabilities of picking up and dropping objects. $P_p(o_i)$ and $P_d(o_i)$ are compared with a random real number $p(p \in [0,1])$ and the comparing results determine whether the object $o_i$ should be picked up or dropped. A high-level description of the LF algorithm is presented in Table 1.

A number of modifications have been introduced to the basic LF algorithm to improve clustering quality and convergence speed [25–29,31,32,39]. But some modifications make the algorithm more complex in parameters. In the following sections, we will show how Renyi entropy is used to create a novel ant movement model.

## 3. Renyi entropy, KECA and their applications in clustering

### 3.1. Renyi entropy and its application in clustering

Alfred Renyi [42] proposed in the 1960s a new information measure, which became known as Renyi's entropy (written as Renyi entropy for simplicity in the paper). For a stochastic variable $Z$ with a probability density function (pdf) $f_Z$, its Renyi entropy is

$$H_R(Z) = \frac{1}{1-\alpha} \log \int f_Z^{\alpha} dz, \quad \alpha > 0, \ \alpha \ne 1. \qquad (5)$$

If $\alpha \to 1$, there is a well relation between Renyi entropy and Shannon's entropy $H_S$:

$$\lim_{\alpha \to 1} H_R(Z) = H_S. \qquad (6)$$

Compared to Shannon's entropy, Renyi entropy provided an easier nonparametric estimator for entropy. Detailed analysis on the relation between Shannon's entropy and Renyi entropy has been described [43].

If $\alpha = 2$, (5) becomes

$$H_R(Z) = -\log \int f_Z^2 dz, \qquad (7)$$

which is called as Renyi's quadratic entropy.

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022

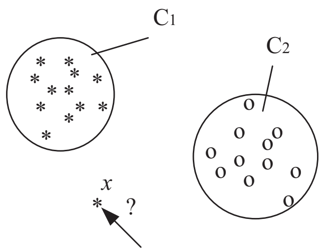


**Fig. 2.** Assigning an object to a cluster.

Let $\{\mathbf{z}\}$ ($\mathbf{z}_i \in R^M$, $i = 1, 2, \ldots, N$), be a set of samples from the variable $Z$ in $M$-dimensional space and assume that the data points in $\{\mathbf{z}\}$ are drawn from the pdf $f_Z$. Renyi quadratic entropy of $\{\mathbf{z}\}$ is written as $H(\{\mathbf{z}\})$. The calculation of $H(\{\mathbf{z}\})$ needs to estimate pdf $f_Z$. A sample-based estimator of Renyi quadratic entropy can be obtained by replacing the actual pdf $f_Z$ by the Parzen window estimator

$$\hat{f}_Z = \frac{1}{N}\sum_{z_i \in \{z\}} W_\sigma(\mathbf{z}, \mathbf{z_i}), \tag{8}$$

where $N$ is the number of data points in $\{\mathbf{z}\}$, $W_\sigma(\cdot, \cdot)$ is the Parzen window. The scale parameter $\sigma$ governs the width of the Parzen window.

Note that the quantity $V(\{z\}) = \int f_Z^2 dz$ may be expressed as $V(\{z\}) = E_f\{f_Z\}$, where $E_f\{\cdot\}$ denotes expectation with respect to the density $f_Z$. By approximating the expectation operator by the sample mean, an estimator for $V(\{z\})$ may be defined as

$$\hat{V}(\{\mathbf{z}\}) = \frac{1}{N}\sum_{z_i \in \{z\}} \hat{f}_Z = \frac{1}{N}\sum_{z_j \in \{z\}} W_\sigma(\mathbf{z}_i, \mathbf{z}_j) = \frac{1}{N^2}\sum_{z_i \in \{z\}}\sum_{z_j \in \{z\}} W_\sigma(\mathbf{z}_i, \mathbf{z}_j)$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_\sigma(\mathbf{z}_i, \mathbf{z}_j). \tag{9}$$

Typically, the Parzen window with a Gaussian kernel function is used [41,44].

$$W_\sigma(\mathbf{z}, \mathbf{z_i}) = G(\mathbf{z} - \mathbf{z_i}, \sigma^2\mathbf{I}), \tag{10}$$

where a symmetric Gaussian kernel with covariance matrix $\sum = \sigma^2\mathbf{I}$ is used

$$G(\mathbf{z}, \sigma^2\mathbf{I}) = \frac{1}{(2\pi)^{M/2}\sigma^M}\exp\left(\frac{-\mathbf{z}^T\mathbf{z}}{2\sigma^2}\right), \tag{11}$$

where $M$ is the dimension number of $\mathbf{z}$. By substituting (10) into (9) and utilizing the properties of the Gaussian kernel, Renyi entropy of $\{\mathbf{z}\}$ is obtained as

$$H(\{\mathbf{z}\}) = -\log\hat{V}(\{\mathbf{z}\}), \tag{12}$$

where

$$\hat{V}(\{\mathbf{z}\}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(\mathbf{z}_i - \mathbf{z}_j, 2\sigma^2\mathbf{I}). \tag{13}$$

If $\{\mathbf{z}\}$ is a cluster, $H(\{\mathbf{z}\})$ can be referred as the within-cluster entropy because it is calculated based on points belonging to the same cluster. Based on this definition, Jenssen [41] introduced a differential entropy clustering. Consider the situation depicted in Fig. 2, they proposed to cluster the object $x$ based on a simple observation. If $x$ is wrongly assigned to $C_2$, the uncertainty or the entropy of $C_2$ will increase more than that of $C_1$ will. Hence, if there are initial clusters $C_k (k = 1, \cdot\cdot\cdot, K)$, assign $x$ to cluster $C_i$ if

$$H(C_i + x) - H(C_i) \le H(C_k + x) - H(C_k), \tag{14}$$

where $H(C_k)$ denotes the entropy of cluster $C_k (k = 1, \cdot\cdot\cdot, K$ and $k \neq i)$.

Compared to Shannon's entropy, the advantage of differential Renyi entropy in clustering is that the entropy is estimated directly from the data samples without imposing assumptions about the pdf. Shannon's definition of entropy (the sum of terms which are weighted logarithms of probability) is not amenable to simple estimation, while Renyi entropy (the sum of the power of probability) is much easier to estimate. This merit enlightened us to use Renyi entropy, rather than Shannon's entropy, to create a novel ant movement model, which will be described in Section 4.

### 3.2. Kernel entropy component analysis

Data transformation is important in data analysis. For the high-dimensional data, data transformation can transform them into an alternative and typically lower dimension, which may reveal the underlying structure of the data. Then further pattern analysis can be performed.

The most well known method of data transformation is principal component analysis (PCA) [45], which is based on the data correlation matrix. It is a linear method ensuring that the transformed data are uncorrelated and preserve maximally the second order statistics of the original data. Another very popular method is Kernel PCA (KPCA) [46]. KPCA performs traditional PCA in a kernel feature space, which is nonlinearly related to the input space. Recently, a new method called Kernel Entropy Component Analysis (KECA) was proposed [47]. KECA is a data transformation that reveals the structure related to Renyi entropy of the input data. Compared to KPCA, KECA is not corresponding to the top eigenvalues and eigenvectors of the kernel matrix. Indeed, KECA typically produces a transformed dataset with a distinct angular structure, thus reveals the cluster structure and information about the underlying labels of the data. Based on KECA, a new spectral clustering algorithm has been proposed [47].

## 4. A novel ant-based clustering algorithm using Renyi entropy

### 4.1. Projection of the objects based on KECA

In ant-based clustering algorithms, the objects are randomly projected onto the plane. As a result, that one pattern corresponds randomly with a pair of coordinates in the plane. This random projection leads to few similarities between the objects in the local neighborhood at the beginning of the algorithm. Therefore, the objects are easily picked up but not easily dropped by the ants. It takes a long time for an object to be similar to nearby objects from the inception of the algorithm.

To reduce the influence of randomness in this stage, we have suggested the modified projection based on PCA [48] and KPCA [49], and the results show they are effective to improve the algorithm's efficiency. In this paper, we applied KECA to replace PCA and KPCA. As the proposer Jessen mentioned, KECA can extract features with a distinct angular structure, thus reveal cluster structure and information about the underlying labels of the data [47]. After the first two Kernel Entropy Components (KECs) are obtained, they need to be processed as the projection coordinates. The process method is similar to that applying PCA and KPCA [48,49], which will not be described here.

### 4.2. The ant movement model using Renyi entropy

We propose a novel ant movement model governed by Renyi entropy. It is shown in Fig. 3.

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022

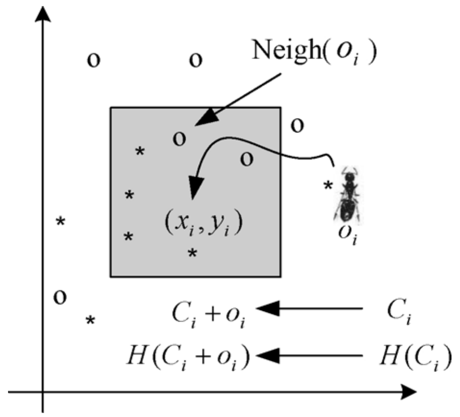


**Fig. 3.** The change of Renyi entropy in *Neigh*($o_i$).

In this model, each object is taken as an ant. Assume that the object (ant) $o_i$ is located at the point ($x_i$,$y_i$) at cycle $t$. Its local neighborhood is signed as *Neigh*($o_i$). When $o_i$ moves to a new place, the entropy of the dataset composed by the objects in *Neigh*($o_i$) will be changed. Suppose $C_i$ is the dataset composed by all objects in *Neigh*($o_i$) before $o_i$ joins. $C_i + o_i$ is the dataset after $o_i$ joins. Then the differential Renyi entropy is

$$\Delta H(C_i) = H(C_i + o_i) - H(C_i). \tag{15}$$

Let

$$\delta(C_i) = \left| \frac{\Delta H(C_i)}{H(C_i)} \right|. \tag{16}$$

Based on the analysis in Section 3.1, if $o_i$ is similar to other objects in *Neigh*($o_i$), the entropy will be decreased or changed in a small range. Therefore, the object $o_i$ should stay in its location and not move if one of following inequalities is satisfied

$$\Delta H(C_i) < 0, \tag{17}$$

$$\delta(C_i) < \delta_0 \tag{18}$$

where $\delta_0$ is a positive and small real number. Otherwise, the object $o_i$ should move to next place.

For example, in the Fig. 3, there are four objects signed as"*" and two objects signed as "*o*". Suppose the data items represented by the object are (In $C_i$, the "*" objects are firstly listed, then the "*o*" objects. In fact, the order of the objects can be random and has no influence on the results):

$$C_i = \begin{bmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 4.6 & 3.1 & 1.5 & 0.2 \\ 6.3 & 2.9 & 5.6 & 1.8 \\ 6.5 & 3.0 & 5.8 & 2.2 \end{bmatrix}, \quad C_i + o_i =$$

$$\begin{bmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 4.6 & 3.1 & 1.5 & 0.2 \\ 6.3 & 2.9 & 5.6 & 1.8 \\ 6.5 & 3.0 & 5.8 & 2.2 \\ 4.6 & 3.1 & 1.3 & 0.2 \end{bmatrix}.$$

Let $\sigma = 0.8$, the entropy and its change can be calculated as $H(C_i) = 4.9651$, $H(C_i + o_i) = 4.8856$, $\Delta H(C_i) = -0.0795$, $\delta(C_i) = 0.016$. From the results, we can know the Renyi entropy of the local neighborhood is decreased after the object $o_i$ joined in, so $o_i$ should stay in this place and not move in this cycle.

The novel ant movement model has following features:

- The local similarity of the object is measured by entropy rather than distance in the old model.
- Ant's movement is governed by the change of Renyi entropy in its local neighborhood.
- The model eliminates the parameter $\alpha$ and the probability conversion functions in LF algorithm, which makes the algorithm simple.

**Table 2**
The pseudo-code of the NAC-RE algorithm.

```
/*initialization*/
All objects are placed on the toroidal grid based on KECA
Initialize all parameters: $r$, $t_{max}$, $\delta_0$
/* main loop*/
for t = 1 to $t_{max}$ do
    for $i$ = 1 to $N$ do ($N$ is the number of the objects(ants))
        find all objects in $o_i$'s local neighborhood
        compute $H(C_i + o_i)$ and $H(C_i)$
        if (17) or (18) is satisfied
            the object(ant) $o_i$ doesn't move
        else
            the object(ant) $o_i$ move to next place in Neigh($o_i$)
        end if
    end for
    Adjust $r$ and $\delta_0$
    $t \leftarrow t + 1$;
end for
Output locations of all objects;
```

Because Renyi entropy can be computed directly by the data samples, it avoids the assumptions when using Shannon's entropy. For example, the attributes of the objects need not be independent. Moreover, the Shannon's entropy needs discretization. For example, if the object's $i$th attribute has value $x$, we have to calculate the sample probability of $x$ in $X_i$ (where $X_i$ is the set of possible discrete values for the $i$th attribute). If there are $n$ attributes, the statistical analysis has to be done $n$ times. Renyi entropy avoids this problem.

### 4.3. The process of the NAC-RE algorithm

Based on the model in Section 4.2, a novel ant-based clustering algorithm using Renyi Entropy (NAC-RE) is proposed. The high level description of the algorithm is shown in Table 2.

It should be noted that for a dataset, $W_\sigma(z_i, z_j)$ in (9) can be calculated beforehand and stored in the matrix $[W_{ij}]$ once $\sigma$ is given. This means that the following calculations are simple matrix manipulations, which saves the algorithm's time greatly.

There are several parameters in the NAC-RE algorithm. The most important two parameters, the radius $r$ and the kernel width $\sigma$ will be discussed in Section 6. Other parameters are set as follows: The size of the projecting plane is set as $N \times N$, where $N$ is the size of the dataset; The maximum cycle number are set as 10,000–50,000; The threshold value $\delta_0$ indicates the permitted degree with which $H$ changes after the object moves to a new place. In our experiments, the initial value of $\delta_0$ is 0.08. With the continuance of clustering, the similar objects will be clustered together. Therefore, the value of $\delta_0$ can be decreased accordingly.

$$\delta_0(t+1) = \begin{cases} 0.95\,\delta_0(t) & \text{if}\, Mod(t, 200) = 0 \\ \delta_0(t) & \text{otherwise} \end{cases} \tag{19}$$

## 5. The theoretical analysis of the NAC-RE algorithm using kernel method

In the essence, the local similarity of the object is measured by Renyi entropy in NAC-RE rather than Euclidean distance or Cosine distance in the old model. In this section, we will show that entropy

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022

metric is feasible and superior to distance metric from the kernel's point of view.

### *5.1. The relation between Renyi entropy and the mean vector in the kernel feature space*

Kernel-based clustering is proposed to cluster some datasets with nonlinear separable classes. It transforms the data into a high-dimensional feature space and performs the clustering in this feature space [50].

Consider a smooth, continuous nonlinear mapping $\Phi$ from the data space to the feature space $F$:

$$\Phi : R^N \to F.$$

Then, the data samples in the input space $z_i \in R^N (i = 1, 2, \ldots, N)$ are mapped into $\Phi(z_1), \Phi(z_2), \ldots, \Phi(z_N)$. Note that the dot production in the feature space can be computed using Mercer kernel in the input space:

$$K(z_i, z_j) = < \Phi(z_i), \Phi(z_j) > . \tag{20}$$

In other words, by employing a specific kernel function, the dot product that it returns implicitly defines the nonlinear mapping $\Phi$ to the feature space [50].

Let us assume that the Parzen window is a positive semi-definite kernel function, for example, the Gaussian Parzen window in Section 3 [51]. Then the Parzen window obeys Mercer's conditions. Hence, an inner-product in kernel induced feature space can be computed

$$W_\sigma(\cdot, \cdot) = K(\cdot, \cdot) = < \Phi(\cdot), \Phi(\cdot) > . \tag{21}$$

Then it is possible to interpret the Renyi entropy-based information theoretic measures in terms of Mercer kernel feature space [51]. The estimator for $V(\{z\})$ in (9) can be re-examined by

$$\hat{V}(\{\mathbf{z}\}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_\sigma(\mathbf{z}_i, \mathbf{z}_j) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} < \Phi(\mathbf{z}_i), \Phi(\mathbf{z}_j) >$$
$$= \left\langle \frac{1}{N}\sum_{i=1}^{N}\Phi(\mathbf{z}_i), \frac{1}{N}\sum_{j=1}^{N}\Phi(\mathbf{z}_j) \right\rangle = < \mathbf{m}, \mathbf{m} > = \|\mathbf{m}\|^2, \tag{22}$$

where $\mathbf{m} = (1/N)\sum_{i=1}^{N}\Phi(\mathbf{z}_i)$ is the mean vector of the data points in $\{z\}$ after mapping to the feature space. Then the Parzen window-based estimator for the Renyi entropy is

$$H(\{\mathbf{z}\}) = -\log \hat{V}(\{\mathbf{z}\}) = -\log \|\mathbf{m}\|^2. \tag{23}$$

This means that the Renyi entropy of $\{\mathbf{z}\}$ equals minus the log of the squared norm of the mean vector of all items in the kernel feature space. The relation indicated by (23) will be used in next section to get the relation between Renyi entropy and the mean distance of dataset in the kernel space.

### *5.2. The relation between Renyi entropy and the mean distance of the kernel space*

Suppose data samples $\mathbf{z}_i$ and $\mathbf{z}_j$ in $\{\mathbf{z}\}$, the Euclidean distance in the input space is

$$d(\mathbf{z}_i, \mathbf{z}_j) = \sqrt{\left\|\mathbf{z}_i - \mathbf{z}_j\right\|^2}, \tag{24}$$

After the samples are mapped into the feature space, the Euclidean distance between $\Phi(\mathbf{z}_i)$ and $\Phi(\mathbf{z}_j)$ in the feature space becomes

$$d_F(\mathbf{z}_i, \mathbf{z}_j) = \sqrt{\left\|\Phi(\mathbf{z}_i) - \Phi(\mathbf{z}_j)\right\|^2}$$
$$= \sqrt{\Phi(\mathbf{z}_i)\cdot\Phi(\mathbf{z}_i) - 2\Phi(\mathbf{z}_i)\cdot\Phi(\mathbf{z}_j) + \Phi(\mathbf{z}_j)\cdot\Phi(\mathbf{z}_j)}. \tag{25}$$

**Table 3**
The change of $\bar{d}_F$ and $H$ for the sample in Fig. 3.

| Item | The dataset | | The changing trend |
|---|---|---|---|
| | $C_i$ | $C_i + o_i$ | |
| $H$ | 4.9651 | 4.8856 | ↓ |
| $\bar{d}_F$ | 0.0173 | 0.0158 | ↓ |

As for all data items in the dataset $\{\mathbf{z}\}$, we define the average distance of $\{\mathbf{z}\}$ in the feature space is $\bar{d}_F = (1/N^2)\sum_{i=1}^{N}\sum_{j=1}^{N} d_F(\mathbf{z}_i, \mathbf{z}_j)^2$. Based on (20), if standard Gaussian kernel is used, then

$$K(\mathbf{z}_i, \mathbf{z}_j) = G(\mathbf{z}_i - \mathbf{z}_j, \sigma^2\mathbf{I}) = \exp\left(\frac{\left\|\mathbf{z}_i - \mathbf{z}_j\right\|^2}{2\sigma^2}\right). \tag{26}$$

$\bar{d}_F$ can be computed as

$$\bar{d}_F = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} d_F(\mathbf{z}_i, \mathbf{z}_j)^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(K(\mathbf{z}_i, \mathbf{z}_i) - 2K(\mathbf{z}_i, \mathbf{z}_j)$$
$$+ K(\mathbf{z}_j, \mathbf{z}_j)) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(2 - 2K(\mathbf{z}_i, \mathbf{z}_j))$$
$$= 2(1 - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}K(\mathbf{z}_i, \mathbf{z}_j)) = 2(1 - \|\mathbf{m}\|^2). \tag{27}$$

Through (23), then we can get the relationship of $\bar{d}_F$, $H$ and $\mathbf{m}$ (here $H(\{z\})$ is written as $H$ to show concisely).

$$\bar{d}_F = 2(1 - \|\mathbf{m}\|^2) = 2(1 - e^{-H}). \tag{28}$$

Or Q2

$$H = -\log\left(1 - \frac{\bar{d}_F}{2}\right). \tag{29}$$

It can be seen that $\bar{d}_F$ is proportional to $H$. Therefore, $H \downarrow \Rightarrow \bar{d}_F \downarrow$

The means of (17) can be re-explained that the mean distance of all objects in $Neigh(o_i)$ in the feature space is decreased when $o_i$ joins its neighborhood. It has been well known that applying the distance in the feature space is more reasonable than applying that in the input space, because non-linearly separable objects in the input space may be linearly separable in the feature space. This is the basic principle of kernel-based clustering. The applications of kernel in K-means, fuzzy K-means, have been proved to be effective to improve the clustering performance [50]. Therefore, the relation gotten by (29) provides the theoretic support for our method using Renyi entropy in ant-based clustering. In Section 6, we will show the distance comparison between the input space and the feature space.

As for the sample in Section 4.2, the changes of $\bar{d}_F$ and $H$ are listed in Table 3.

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022

**Table 4**
The datasets used for assessing all clustering algorithms.

| Name | $C$ | $D$ | $N$ | $N_i$ |
|---|---|---|---|---|
| Square | 4 | 2 | 400 | 100,100,100,100 |
| Ring | 2 | 2 | 200 | 100,100 |
| Line | 2 | 2 | 200 | 100,100 |
| Moon | 2 | 2 | 210 | 105,105 |
| Iris | 3 | 4 | 150 | 50,50,50 |
| Wine | 3 | 13 | 178 | 59,71,48 |
| Wisconsin | 2 | 9 | 699 | 458,241 |
| Zoo | 7 | 16 | 101 | 41,20,5,13,4,8,10 |

## 6. Experimental results and analysis

### 6.1. The experimental condition

#### 6.1.1. Evaluation functions

The following functions are used to evaluate the performance of the NAC-RE algorithm and other clustering algorithms.

(1) The F-measure ($F$)
(2) The Dunn Index ($DI$)
(3) The Error Rate ($ER$)
(4) Time cost ($T$)

$F$ and $DI$ are to be maximized while $ER$ and $T$ are to be minimized [29].

#### 6.1.2. Experimental data

Eight datasets, four synthetic and four real, are used for assessing the algorithms. Some of these datasets are benchmarks and widely applied in ant-based clustering. The datasets are briefly introduced in Table 4 ($C$ is the number of the clusters, $D$ is the dimensionality, $N$ is the total number of the data items, $N_i$ is the number of items of cluster $i$).

*Square*: The Square dataset has been used in many ant-based clustering algorithms. The dataset is two-dimensional and consists of four clusters arranged as a square. The data are generated according to a normal distribution $N(u, \sigma^2)$. The normal distributions of the four clusters in our study are as follows: $(N(-5, 2), N(-5, 2))$, $(N(5, 2), N(5, 2))$, $(N(-5, 2), N(5, 2))$, and $(N(5, 2), N(-5, 2))$.

*Ring*: This dataset is generated by two distributions: an isotropic Gaussian and a uniform "Ring" distribution. A total of 100 data points were drawn for each distribution.

*Line*: The dataset is composed by two clusters. One is Gaussian and one is linear with two parts.

*Moon*: The dataset includes two parts of data with a valley structure, which is often used for testing some clustering algorithms such as spectral clustering and manifold clustering.

The real datasets Iris, Wine, Wisconsin and Zoo, are all from the database of UCI for machine learning [52]. These datasets are often used for testing the performance of all kinds of algorithms.

#### 6.1.3. The comparison clustering algorithms

The NAC-RE algorithm was compared with the following algorithms:

- The classic clustering algorithm
- ○ K-means
- The classic kernel-based algorithm
- ○ Kernel-based K-means (KK-means)
- The ant-based clustering algorithms
  - ○ LF Algorithm [21]
  - ○ ATTA [29], which represents the latest modified algorithm of ant-based clustering.
  - ○ ACK [49], which is ant-based clustering algorithm integrated with kernel method and proposed by us.
  - ○ NAC-RE-K: The algorithm is designed for comparing KPCA and KECA. The initial projection of the objects is modified by KPCA, and the ant's clustering part is the same as that in NAC-RE.

Another algorithm ACAM is not included here because its modifications are complex. We cannot program ACAM algorithm by our codes. Its experimental results on published papers are not sufficient for our comparison. ATTA can be as the example of the latest ant-based clustering and its performance is superior to ACAM. The program of ATTA can be downloaded from the author's website [53]. ATTA is programmed in C++ and executed in the Linux operating system. LF and ACK are programmed in Matlab and performed on an Intel core E7200 2.53 GHz personal computer. All presented results by evaluation functions have been averaged over 10 runs.

### 6.2. The simplification of NAC-RE in the parameters

In this section, we will indicate that NAC-RE has simpler parameters than other ant-based clustering algorithms and discuss how the parameters are selected.

Table 5 shows the comparison of the number of the parameters in different algorithms. K-means is the simplest. Compared with K-means and KK-means, ant-based clustering algorithms are relative complex. Among ant-based clustering, the number of parameters in NAC-RE is the smallest, which prevents that complex parameters setting affects the clustering results. The modifications in ATTA have improved the clustering quality, but at the same time, the complexity of the algorithm has been increased. For example, the author Handl has pointed out that the modified threshold functions have been experimentally derived [29]. Moreover, by applying entropy metric, NAC-RE eliminates the similarity parameter $\alpha$, which is difficult to be adjusted properly [29–31,39].

Table 5 also shows the comparison of algorithms in time complexity($N$ is the total number of the data items, $n_{ant}$ is the number of ants, $K$ is the number of clusters, $t_{\max}$ is the maximum iteration number). The time complexity of the algorithm is mainly affected by the number of iteration. Commonly, tens of thousands of iterations are needed for LF and ATTA [17,29,39]. For ACK, NAC-RE-K and NAC-RE, their time complexity is relative to the number of data

**Table 5**
The comparison of the parameters in several algorithms.

| Algorithm | Parameters | | | Time complexity |
|---|---|---|---|---|
| | Adjusted | Constant | Number | |
| K-means | – | $K$ | 1 | $O(N \cdot K \cdot t_{\max})$ |
| KK-means | – | $K$, $\sigma$ | 2 | $O(N \cdot K \cdot t_{\max})$ |
| LF | $r$,$\alpha$ | $n$,$t_{\max}$,$k_1$,$k_2$ | 6 | $O(n_{ant} \cdot t_{\max})$ |
| ATTA | $r$,$\alpha$ | $n$,$t_{\max}$,$t_{start}$,$t_{end}$ *stepsize*, *memorysize* | 8 | $O(n_{ant} \cdot t_{\max})$ |
| ACK | $r$,$\alpha$,$\beta$ | $t_{\max}$,$K_1$,$\sigma$ | 6 | $O(N \cdot t_{\max})$ |
| NAC-RE-K | $r$,$\delta_0$ | $t_{\max}$,$\sigma$ | 4 | $O(N \cdot t_{\max})$ |
| NAC-RE | $r$,$\delta_0$ | $t_{\max}$,$\sigma$ | 4 | $O(N \cdot t_{\max})$ |

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022

**Table 6**
The comparison of NAC-RE with non ant-based clustering algorithms.

| Dataset | Evaluation parameter | Algorithms | | |
|---|---|---|---|---|
| | | K-means | KK-means | NAC-RE |
| Square | *F* | 0.983 | 0.984 | 0.981 |
| | *DI* | 3.702 | 3.717 | 3.772 |
| | *ER* | 0.99 | 0.99 | 1.05 |
| | *T* | 25.16 | 28.78 | 210.26 |
| Ring | *F* | 0.664 | 0.893 | 0.880 |
| | *DI* | 1.349 | 1.714 | 1.897 |
| | *ER* | 19.65 | 7.02 | 3.94 |
| | *T* | 25.26 | 27.69 | 90.83 |
| Line | *F* | 0.637 | 0.818 | 0.814 |
| | *DI* | 1.252 | 1.297 | 1.395 |
| | *ER* | 15.45 | 5.54 | 4.93 |
| | *T* | 23.35 | 23.46 | 134.66 |
| Moon | *F* | 0.794 | 0.890 | 0.882 |
| | *DI* | 2.165 | 2.454 | 2.469 |
| | *ER* | 9.05 | 6.72 | 6.92 |
| | *T* | 30.10 | 30.12 | 92.84 |
| Iris | *F* | 0.822 | 0.830 | 0.815 |
| | *DI* | 2.669 | 2.686 | 2.746 |
| | *ER* | 10.72 | 8.02 | 11.84 |
| | *T* | 30.75 | 31.66 | 91.28 |
| Wine | *F* | 0.813 | 0.859 | 0.862 |
| | *DI* | 1.932 | 3.927 | 4.209 |
| | *ER* | 5.24 | 3.14 | 3.14 |
| | *T* | 30.10 | 31.25 | 90.36 |
| WI | *F* | 0.956 | 0.960 | 0.980 |
| | *DI* | 4.155 | 4.872 | 5.454 |
| | *ER* | 4.24 | 4.20 | 3.05 |
| | *T* | 33.64 | 46.55 | 203.69 |
| Zoo | *F* | 0.774 | 0.804 | 0.816 |
| | *DI* | 1.284 | 1.386 | 1.446 |
| | *ER* | 23.40 | 14.39 | 13.02 |
| | *T* | 10.58 | 10.69 | 76.36 |

item because each item is taken as an ant. The $t_{\max}$ in these three algorithms is far less than that in LF and ATTA because of modified projection. The detailed comparison of time cost will be shown in Section 6.3.

There are three important parameters in NAC-RE, which are simple and easily controlled. The parameter $\delta_0$ has been discussed in Section 4.3, other two parameters *r* and $\sigma$ are discussed here.

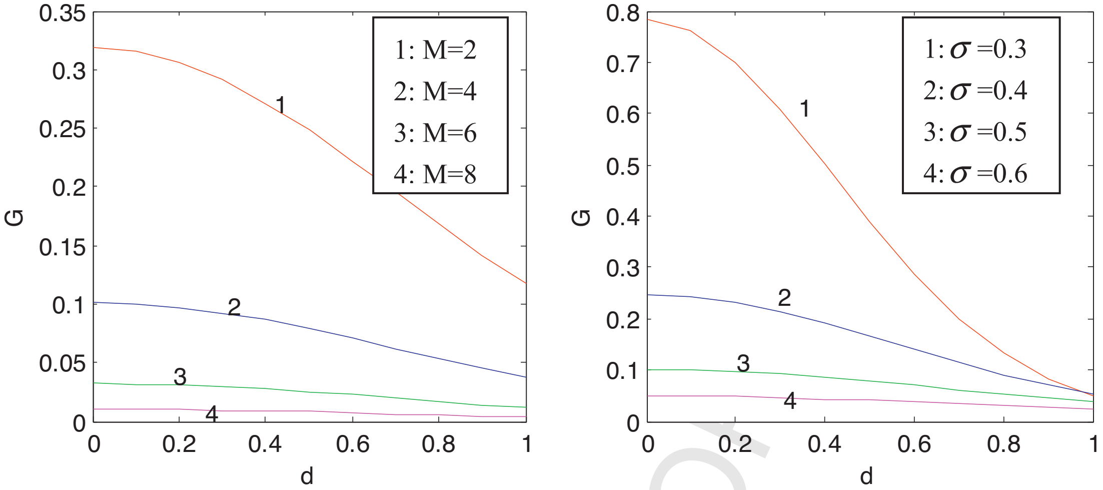
- The radius *r*
  The radius *r* determines the region that the ant perceives. A larger radius means that it takes in more information but there is a higher time cost. Furthermore, a larger radius inhibits the quick formation of clusters in the initial phase. We applied a changing radius that gradually increases over cycles [27–29,39]. The initial value $r_1$ is 1, and the maximum $r_{\max}$ is 5.
- The kernel size $\sigma$
  The kernel size $\sigma$ is another important parameter because it determines the width of the Parzen window. How to set $\sigma$ is an important research focus in the applications of Gauss function in Support Vector Machine (SVM) and other kernel-based methods [54]. There are no good ways to solve this problem. Cross-validation and the leave-one-out technique are usually used. Sometimes, $\sigma$ can be selected by experience. In this paper, we applied the cut and trial method, and the value that generated a good result in KECA was selected.

A general guideline to choose $\sigma$ can be given by analyzing the Gauss function $G(\mathbf{z}_i - \mathbf{z}_j, 2\sigma^2\mathbf{I})$ in (12). Assume the distance $\mathbf{z}_i - \mathbf{z}_j$ is indicated as *d* and processed in the range of [0 1]. Fig. 4(a) shows the curves of Gauss function when $\sigma$ is constant and *M* is changing (*M* is the dimension number of $\mathbf{z}$). The less *M* is, the more obviously that curve declines. Fig. 4(b) shows the curves of Gauss function when *M* is constant and $\sigma$ is changing. The less $\sigma$ is, the more obviously

**Table 7**
The comparison of NAC-RE with ant-based clustering algorithms.

| Dataset | Evaluation parameter | Algorithms | | | | |
|---|---|---|---|---|---|---|
| | | LF | ATTA | ACK | NAC-RE-K | NAC-RE |
| Square | *F* | 0.894 | 0.980 | 0.989 | 0.983 | 0.981 |
| | *DI* | 2.896 | 3.654 | 3.926 | 3.824 | 3.772 |
| | *ER* | 2.05 | 1.02 | 0.97 | 1.01 | 1.05 |
| | *T* | 450.72 | 5.14 | 204.78 | 223.54 | 210.26 |
| Ring | *F* | 0.842 | 0.843 | 0.982 | 0.876 | 0.880 |
| | *DI* | 1.478 | 1.895 | 1.900 | 1.802 | 1.897 |
| | *ER* | 12.67 | 8.48 | 3.06 | 4.64 | 3.94 |
| | *T* | 238.45 | 8.46 | 85.66 | 83.78 | 90.83 |
| Line | *F* | 0.593 | 0.602 | 0.826 | 0.811 | 0.814 |
| | *DI* | 0.973 | 1.058 | 1.489 | 1.320 | 1.395 |
| | *ER* | 17.23 | 15.78 | 5.42 | 5.12 | 4.93 |
| | *T* | 296.46 | 7.97 | 163.26 | 136.27 | 134.66 |
| Moon | *F* | 0.748 | 0.838 | 0.882 | 0.880 | 0.882 |
| | *DI* | 2.044 | 2.466 | 2.453 | 2.454 | 2.469 |
| | *ER* | 11.30 | 9.47 | 7.43 | 7.14 | 6.92 |
| | *T* | 199.51 | 7.62 | 101.22 | 93.38 | 92.84 |
| Iris | *F* | 0.772 | 0.818 | 0.835 | 0.827 | 0.815 |
| | *DI* | 2.118 | 2.923 | 2.898 | 2.670 | 2.746 |
| | *ER* | 14.49 | 12.64 | 7.45 | 9.02 | 11.84 |
| | *T* | 186.15 | 3.30 | 80.42 | 126.42 | 91.28 |
| Wine | *F* | 0.856 | 0.855 | 0.868 | 0.860 | 0.862 |
| | *DI* | 2.034 | 4.242 | 4.197 | 4.022 | 4.209 |
| | *ER* | 3.90 | 3.85 | 3.02 | 3.26 | 3.14 |
| | *T* | 199.51 | 4.25 | 101.22 | 113.49 | 90.36 |
| WI | *F* | 0.874 | 0.968 | 0.972 | 0.968 | 0.980 |
| | *DI* | 4.963 | 5.488 | 5.428 | 5.343 | 5.454 |
| | *ER* | 6.03 | 4.08 | 3.42 | 3.58 | 3.05 |
| | *T* | 356.65 | 10.27 | 252.95 | 237.88 | 203.69 |
| Zoo | *F* | 0.785 | 0.825 | 0.818 | 0.810 | 0.816 |
| | *DI* | 1.147 | 1.396 | 1.562 | 1.339 | 1.446 |
| | *ER* | 21.98 | 11.27 | 12.83 | 14.02 | 13.02 |
| | *T* | 166.65 | 6.62 | 80.95 | 80.44 | 76.36 |

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022



(a) $\sigma$ =0.5

(b) M=4

**Fig. 4.** The effect of $\sigma$ and $M$ on Gauss function.

that curve declines. Gauss function is hoped to be distinct when distances are different, that means the change of entropy ($\Delta H$) will be obvious. In Fig. 4, $\sigma = 0.3$ is proper when $M$ is 4. If $M$ is larger, $\sigma$ can be set a less value.

It should be noted that the differential value and the ratio value of entropy are used in our application. Although $\sigma$ is variant, the changing trends of differential value and the ratio value of entropy are the same. So $\sigma$ can be selected in a range, which does not affect the final clustering results.

### 6.3. The comparison results and discussion

#### 6.3.1. Statistical testing on the comparison results

The comparison results of NAC-RE with non ant-based clustering and ant-based clustering algorithms are shown in Tables 6 and 7.

Statistical testing is used for comparing the algorithms. Demšar examined several statistical tests and studied their suitability for comparison of two or more classifiers over multiple datasets [55]. The Friedman test with the corresponding post hoc tests recommended in [55] is used here.

The Friedman test ranks the algorithm for each data set separately. The best performing algorithm gets the rank of 1, the second best rank 2. In case of ties, average rank is assigned. Let $r_i^j$ be the rank of the $j$th of $k$ algorithms on the $i$th of $N$ data sets. The Friedman test compares the average ranks of algorithm, $R_j = (1/N)\sum_i r_i^j$. Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks $R_j$ should be equal. The Friedman statistic

$$\chi_F^2 = \frac{12 \cdot N}{k \cdot (k+1)} \left[ \sum_j R_j^2 - \frac{k \cdot (k+1)^2}{4} \right] \tag{30}$$

is distributed according to $\chi_F^2$ with $k-1$ degrees of freedom. Friedman's $\chi_F^2$ is undesirably conservative. So another statistic was suggested by Iman and Davenport [56] is

$$F_F = \frac{(N-1) \cdot \chi_F^2}{N \cdot (k-1) - \chi_F^2} \tag{31}$$

It is distributed according to the $F$-distribution with $k-1$ and $(k-1) \cdot (N-1)$ degrees of freedom.

**Table 8**
$\chi_F^2$ and $F_F$ distribution.

| | $\chi_F^2(2)$ | $F_F(2,14)$ | $\chi_F^2(4)$ | $F_F(4,28)$ |
|---|---|---|---|---|
| $\alpha = 0.05$ | 5.991 | 3.74 | 9.488 | 2.71 |
| $\alpha = 0.1$ | 4.605 | 2.73 | 7.779 | 2.16 |

Two cases will be analyzed in the paper, one is comparing three algorithms for eight data sets in Table 6; the other is comparing five algorithms for eight data sets in Table 7. The selected $\chi_F^2$ and $F_F$ distributions are listed in Table 8.

If the null-hypothesis is rejected, a *post hoc* test can proceed. The Nemenyi test is used when all the algorithms are compared to each other. The performance of two clustering algorithms is significantly different if the corresponding average ranks differ by at least the critical difference

$$CD = q_\alpha \sqrt{\frac{k \cdot (k+1)}{6 \cdot N}} \tag{32}$$

When all clustering algorithms are compared with a control clustering algorithm for example, comparing the newly proposed algorithm with several existing methods, Boferroni–Dunn test can replace the Nemenyi test, which is more powerful in this specific case. The CD can be calculated using the same way for the Nemenyi test, but the critical values is different. For convenience, in the cases of the paper (three and five algorithms, eight datasets), the critical values of two tests are listed in Table 9, which are partly selected from the tables in [55]. The corresponding critical differences are listed in Table 10.

#### 6.3.2. Summary of algorithms' performance

The statistical testing is conducted on evaluation parameters. The results are listed in Table 11.

Based on the values of $\chi_F^2$ and $F_F$ (for $DI$ and $T$, $F_F$ is non-existent because the denominator in (31) is null. $\chi_F^2$ distribution can be

**Table 9**
Critical values for the two-tailed Nemenyi test and Bonferroni–Dunn test.

| #algorithms | Nemenyi test | | Bonferroni–Dunn test | |
|---|---|---|---|---|
| | $k = 3$ | $k = 5$ | $k = 3$ | $k = 5$ |
| $\alpha = 0.05$ | 2.343 | 2.728 | 2.241 | 2.498 |
| $\alpha = 0.1$ | 2.052 | 2.459 | 1.960 | 2.241 |

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022

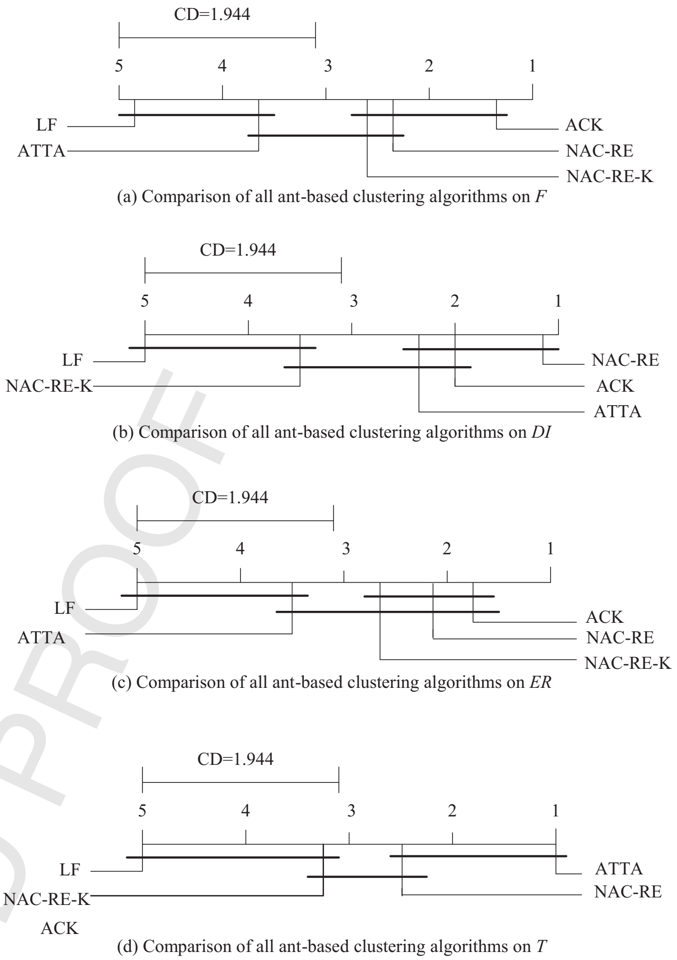
**Table 10**
Critical differences for the two-tailed Nemenyi test and Bonferroni–Dunn test ($N = 8$).

| #algorithms | Nemenyi test | | Bonferroni–Dunn test | |
|---|---|---|---|---|
| | $k = 3$ | $k = 5$ | $k = 3$ | $k = 5$ |
| $\alpha = 0.05$ | 1.172 | 2.156 | 1.121 | 1.249 |
| $\alpha = 0.1$ | 1.026 | 1.944 | 0.980 | 1.121 |

used in this condition), all null-hypotheses are rejected. Therefore the algorithms are different. The Nemenyi test for pairwise comparisons can be conducted. In terms of *F*, the difference between K-means and KK-means 1.375 is larger than critical difference 1.172($\alpha = 0.05$), we can conclude that the performance of KK-means is significantly better than K-means. As for NAC-RE, because its differences with other two algorithms(0.875 and 0.5) are smaller than 1.172(1.026 when $\alpha = 0.1$), so we cannot tell which group NAC-RE belongs to. In terms of *DI*, the best one NAC-RE is obviously different from the worst one K-means. The difference between KK-means with the others is 1, which is less than 1.026($\alpha = 0.1$), so this difference is not significant. In this case, NAC-RE can be looked as a newly proposed algorithm and Boferroni–Dunn test is used to compare it with the other two existing methods. Because 1 is larger than 0.980($\alpha = 0.1$), we can conclude that NAC-RE performs significantly better than existing two ones. The similar analysis can be conducted on terms of *T*, we can see NAC-RE is obviously worse than the others. Based on the average rank, NAC-RE is the best one in *DI* while K-means is in *T*. In terms of *ER*, KK-means and NAC-RE belong to the same group. Their performance is significantly better than K-means.

The general descending order of the algorithms based on average rank is KK-means, NAC-RE and K-means. The improved performance of KK-means shows that the application of kernel is effective. K-means and KK-means are highly efficient compared with NAC-RE. But it should be noted that these two algorithms require a priori knowledge of the number of clusters. In our experiment, they were run by being given the correct number of clusters. While NAC-RE does not need know this priori knowledge, which is an important merit.

The statistical testing results of five ant-based clustering algorithms on eight datasets are listed in Table 12. For convenience, the $\chi^2_F$ and $F_F$ distribution, the Critical Value and the Critical Difference have been computed and listed in Tables 8–10. Using the same analyzing way in Section 6.3.2, we can know that all null-hypotheses are rejected, which means that the algorithms are different in terms of four parameters.



(a) Comparison of all ant-based clustering algorithms on *F*

(b) Comparison of all ant-based clustering algorithms on *DI*

(c) Comparison of all ant-based clustering algorithms on *ER*

(d) Comparison of all ant-based clustering algorithms on *T*

**Fig. 5.** Comparison of all ant-based clustering algorithms with Nemenyi test (groups of algorithms that are not significantly different are connected, $\alpha = 0.1$).

The graphical presentation of results suggested by Demšar [55] is applied here.

The analysis on *F* reveals that LF performs significantly worse than ACK, NAC-RE and NAC-RE-K. ATTA is significantly worse than ACK. The data is not sufficient to conclude whether ATTA performs as LF or the better NAC-RE and NAC-RE-K. The same analyses can be conducted on *DI*, *ER* and *T*. The general description of the comparison of algorithms can be described as:

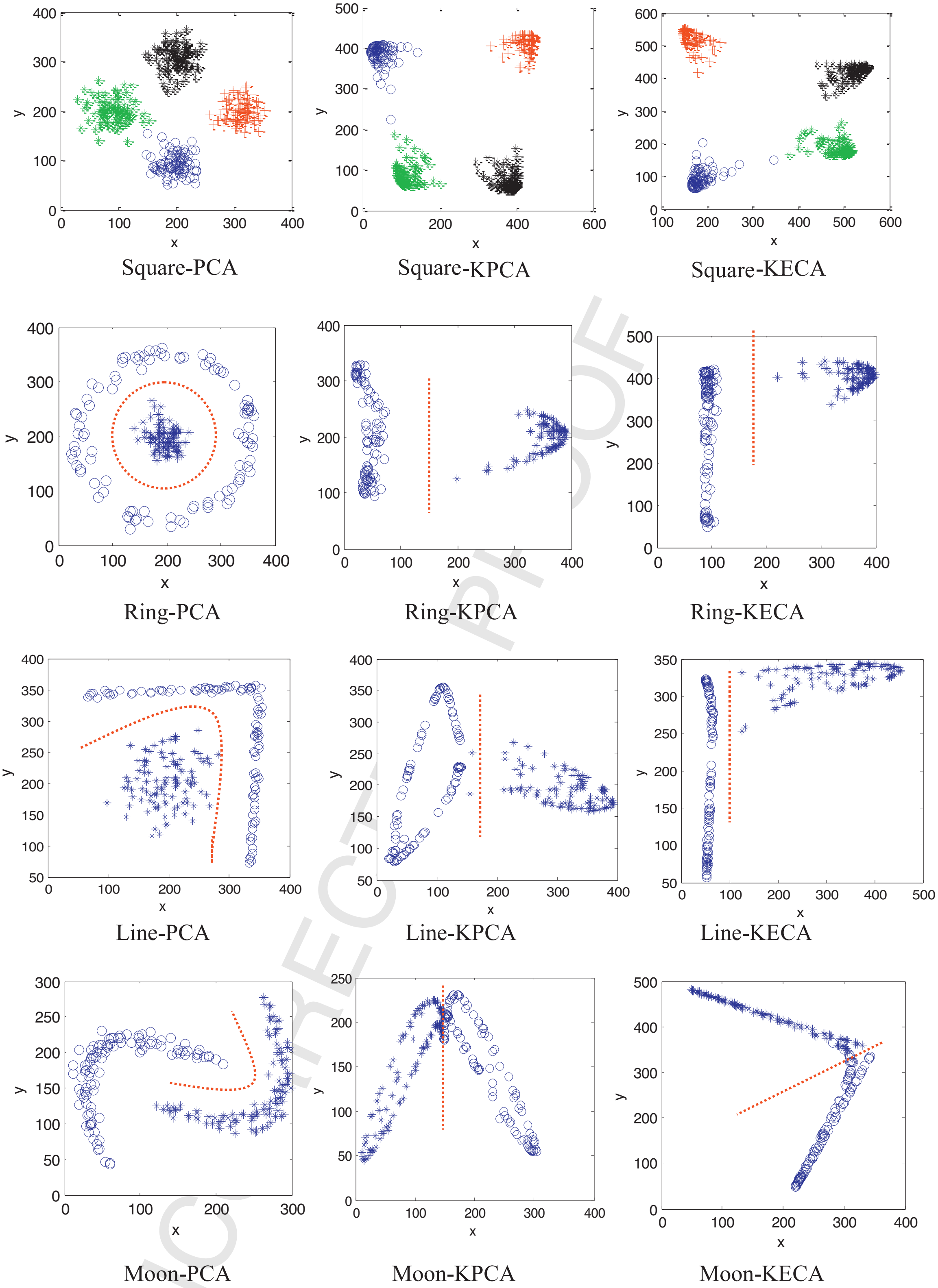
**Table 11**
Q4 The statistical testing of three algorithms.

| Evaluation parameter | Average rank | | | $\chi^2_F$ | $F_F$ |
|---|---|---|---|---|---|
| | K-means | KK-means | NAC-RE | | |
| *F* | 2.75 | **1.375** | 1.875 | 7.75 | 6.576 |
| *DI* | 3 | 2 | **1** | 16 | – |
| *ER* | 2.6875 | **1.625** | 1.6875 | 5.6825 | 3.849 |
| *T* | **1** | 2 | 3 | 16 | – |

**Table 12**
The statistical testing of five algorithms.

| Evaluation parameter | Average rank | | | | | $\chi^2_F$ | $F_F$ |
|---|---|---|---|---|---|---|---|
| | LF | ATTA | ACK | NAC-RE-K | NAC-RE | | |
| *F* | 4.875 | 3.5625 | **1.3125** | 2.6875 | 2.3125 | 18.4 | 9.471 |
| *DI* | 5 | 2.375 | 2 | 3.5 | **1.125** | 10.1 | 3.228 |
| *ER* | 5 | 3.5 | **1.75** | 2.625 | 2.125 | 21.5 | 14.333 |
| *T* | 5 | **1** | 3.25 | 3.25 | 2.5 | 26.8 | 36.077 |

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022



Square-PCA

Square-KPCA

Square-KECA

Ring-PCA

Ring-KPCA

Ring-KECA

Line-PCA

Line-KPCA

Line-KECA

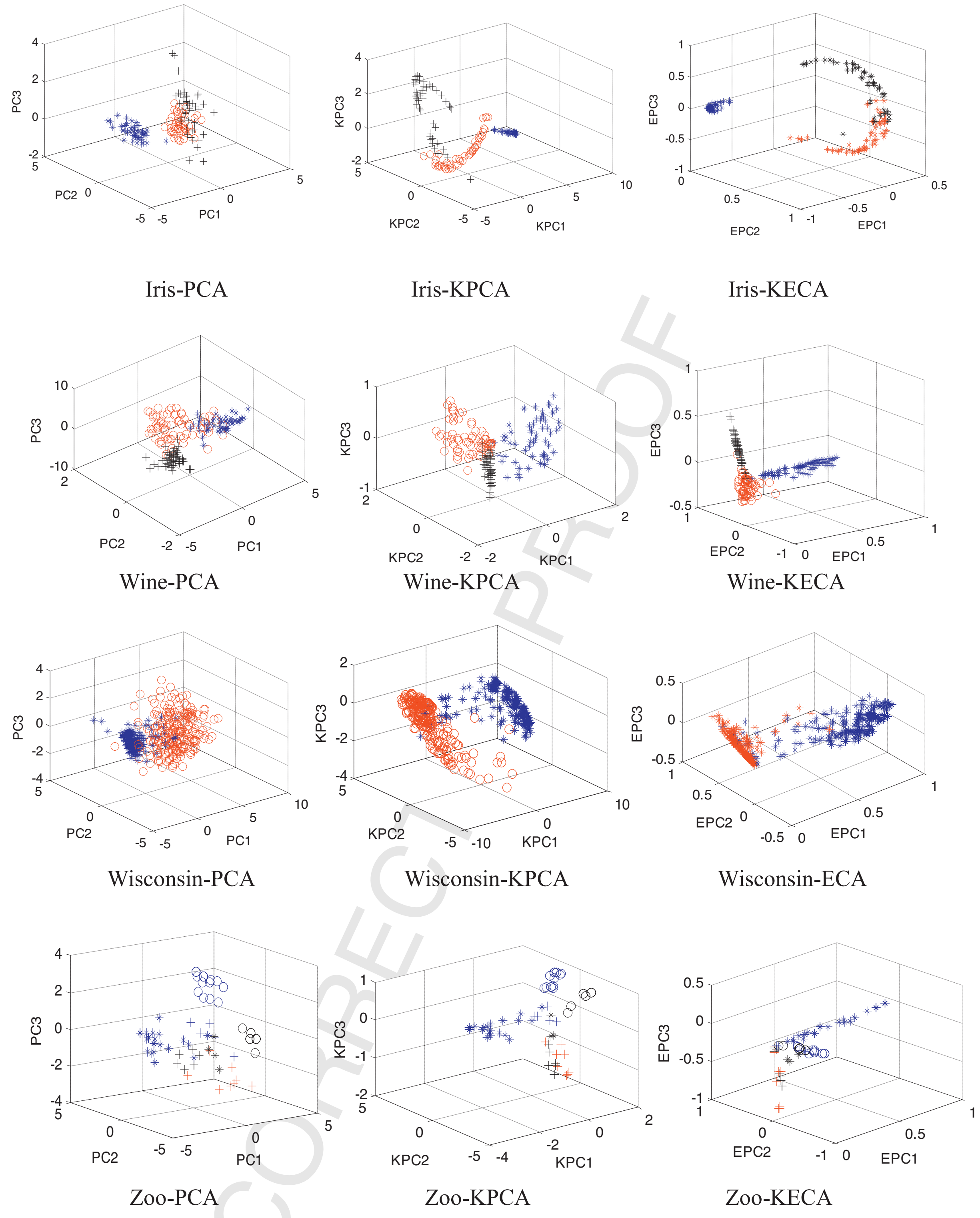
Moon-PCA

Moon-KPCA

Moon-KECA

**Fig. 6.** The projections of synthetic datasets.

- The significant difference of ACK, NAC-RE, NAC-RE-K, compared with LF in terms of *F* and *ER* (the difference is larger than CD) indicate that the application of kernel or entropy is effective.
- NAC-RE and ACK have similar performance, since they are not significantly different in four parameters.
- The significant difference between NAC-RE and NAC-RE-K in *DI* shows that KECA plays an important role in improving the performance in *DI*.
- ACK is significantly better than ATTA in *F* while worse in *T*, which indicates that the clustering accuracy sometimes is at the cost of time. (it should be noted that ATTA codes are written in C++ and



Iris-PCA Iris-KPCA Iris-KECA

Wine-PCA Wine-KPCA Wine-KECA

Wisconsin-PCA Wisconsin-KPCA Wisconsin-ECA

Zoo-PCA Zoo-KPCA Zoo-KECA

**Fig. 7.** The comparisons of three processing methods of real datasets.

run in the Linux operation system, which saves time compared to Matlab codes in the Windows.)

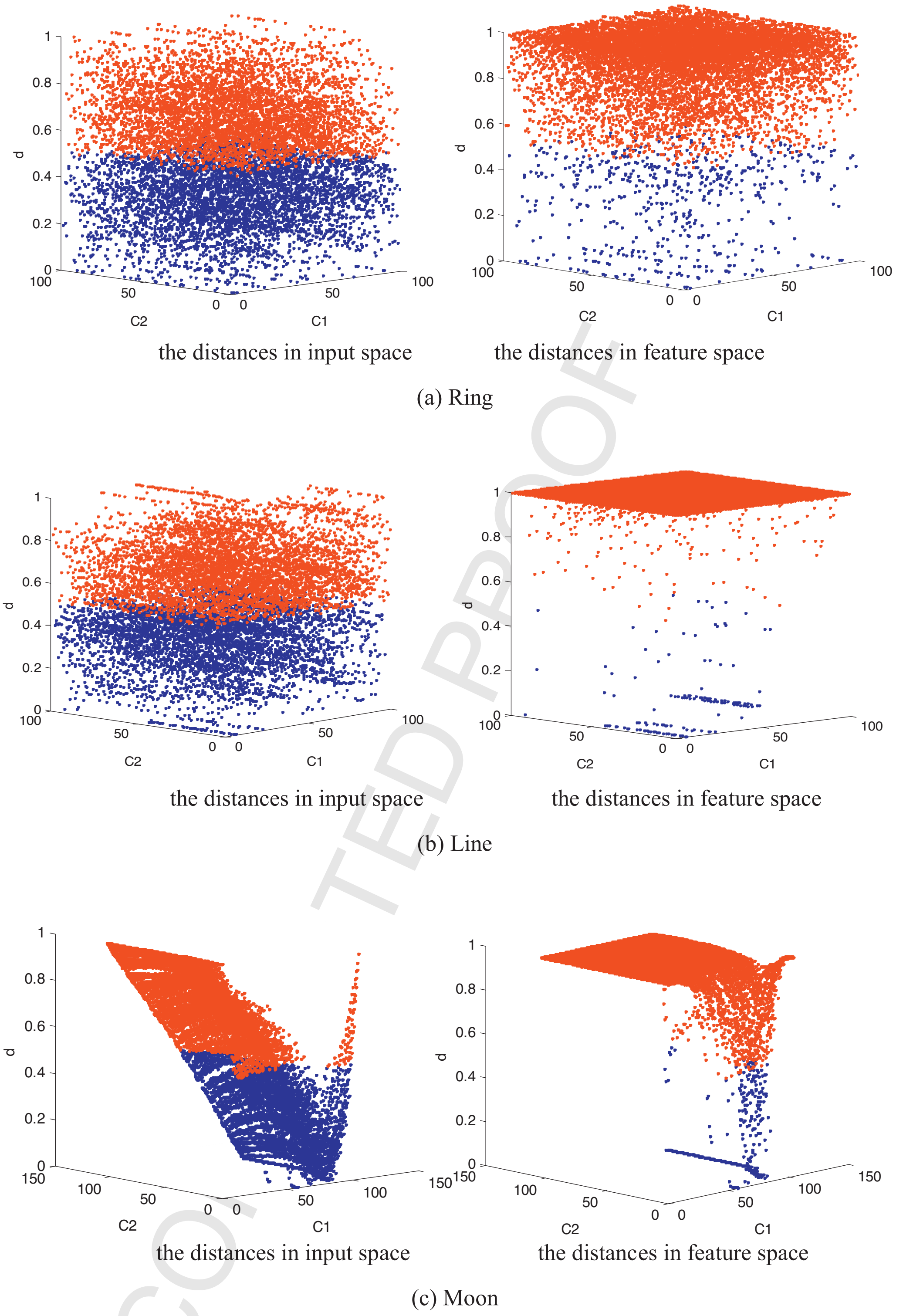
If NAC-RE is seen as a newly algorithm and Boferroni-Dunn test is used to compare it with the others, we can conclude that it is significantly different from LF and ATTA in all terms (the difference is larger than 1.121 in Table 10 when $\alpha = 0.1$). Instead, the difference between NAC-RE and ACK is not obvious in all terms. The difference between NAC-RE and NAC-RE-K is also not obvious except DI.

### 6.3.3. KECA vs. KPCA and PCA

Fig. 6 shows the projections based on PCA, KPCA and KECA of synthetic datasets. Some conclusions can be drawn from Fig. 6.

- The projections based on PCA, KPCA and KECA can create rough clusters, which saves the algorithm's time. Especially for the datasets Square and Ring, the initial projections clearly create clusters, and even no further clustering is required.

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022



(a) Ring

(b) Line

(c) Moon

**Fig. 8.** The comparison of the distances.

- Compared with PCA, KPCA and KECA are superior because the non-linearly separable objects in PCA can be linearly separable in KPCA and KECA, which is shown in Ring, Line and Moon.
- Compared with KPCA, KECA projection shows a distinct angle-based structure. The two clusters in Ring, Line and Moon are almost separated by a 90° angle (the proposer of KECA, Jenssen, had pointed this feature in his research [40]). Moreover, each cluster in KECA is more compacted than KPCA, which is almost compacted to a line.

Because the data items in real datasets are multi-dimensional, to compare the role of PCA, KPCA and KECA, the first three principal components of datasets are shown in Fig. 7.

Some conclusions can be drawn from Fig. 7:

- Iris and Wine are almost linearly separable for three clusters while Wisconsin and Zoo are not linearly separable.
- As for all datasets, KPCA and KECA have superiority over PCA. The clusters are more separable after KPCA and KECA,

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022

especially for dataset Wisconsin. It proves that kernel map is effective.
- Compared with KPCA, the clusters in KECA are distributed in an angle structure. Moreover, each cluster is more compacted in its angle direction.

Comparing NAC-RE-K, NAC-RE, ACK in Table 6, Table 7 and Fig. 5, we can see

- NAC-RE is superior to NAC-RE-K which indicates the initial projection based on KECA can improve not only clustering quality but also efficiency.
- The clustering quality of ACK is better than NAC-RE-K, which indicates the clustering quality is more dependent on the late part of clustering method than the projection.
- The significant difference between NAC-RE and NAC-RE-K in *DI* shows that KECA plays an important role in improving the performance in *DI*.

### 6.3.4. Distance similarity vs. entropy similarity

As for non-linearly separable datasets, such as Ring, Line and Moon, the clustering using the distance measure in the input space cannot get satisfied results. However, the objects will be linearly separable in the feature space after kernel mapping. The improved performance of KK-means to K-means has proved this conclusion. As analyzed in Section 5, the similarity measured by entropy is in essence direct with the distance in kernel space, which means that NAC-RE integrates the merits of kernel method.

Suppose two clusters in datasets Ring, Line and Moon are $C_1$ and $C_2$. The distance between every point in cluster $C_1$ and that in cluster $C_2$ is computed and processed in the range [0,1]. Fig. 8 shows the comparison of all distances of two clusters in input space with those in feature space (to show more clearly, the values larger than 0.5 are shown by red color while the others are shown by blue color). It can be seen from the figures, for each dataset, the distances in the input space smaller than 0.5 are almost equal to those larger than 0.5, which means there are many objects difficult to recognize its true cluster. In contrast, the distances smaller than 0.5 become a very smaller part in the feature space. Especially for dataset Line and Moon, most distance values are approximate to 1, which means the most objects become separable in the feature space. So ant-based clustering using entropy metric can get better clustering results.

## 7. Conclusion

Compared with other clustering algorithms, the most advantage of ant-based clustering algorithms is they do not need any prior knowledge about clustering [19]. The clustering process is organized by ants' behavior [16,19–21]. The clustering results are visible [19,22,27,29]; the algorithm can be performed by parallel computing [25,31].

This paper proposed a novel ant-based clustering algorithm integrated with Renyi entropy (NAC-RE). The algorithm used KECA to modify the random projection of objects, and applied a novel ant movement model governed by Renyi entropy. NAC-RE shows a comparable performance with ACK and ATTA. Because it integrates the merits of kernel method in essence, it can get good results for non-linearly separable datasets. Compared with KPCA, the projection based on KECA can create more compacted clusters, therefore, NAC-RE is time saving. Moreover, NAC-RE is simpler than other ant-based algorithms in parameters. The theoretic analysis and experimental comparison show the novel algorithm is reasonable and effective.

The algorithm gives a novel application of information entropy in ant-based clustering. The implemental style of the algorithm is original. The following items are needed to be further studied:

- How to improve the clustering efficiency by modifying the ant's movement model based on entropy. Although random projection was modified, NAC-RE is not efficient compared with K-means and KK-means. The ant's movement must be optimized. Local optimization methods should be developed.
- How to set the parameters suitably. The setting of $\sigma$ and its effect on clustering results need further study.
- The assessment of NAC-RE using more and large datasets. The algorithm should be evaluated using more datasets. Especially, some complicated datasets with a larger number of objects' attributes should be collected.

## Acknowledgements

This paper is supported by the National Natural Science Foundation of China (No. 50705054) and the key national project of ITER China (No. 2011GB113005). The authors gratefully acknowledge the providers of the datasets in the UCI machine learning database for research. Moreover, the authors sincerely appreciate the reviewers for their advice and suggestions to improve the quality of the paper.

## References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks (1995) 1942–1948.
[2] J. Kennedy, R. Eberhart, A discrete binary version of the particle swarm algorithm, IEEE Iternational Conference on Computational Cybernetics and Simulation 5 (1997) 4104–4108.
[3] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics – Part B 26 (1) (1996) 1–13.
[4] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.
[5] L.F. Giraldo, F. Lozano, N. Quijano, Foraging theory for dimensionality reduction of clustered data, Machine Learning 82 (1) (2011) 71–90.
[6] C. Grosan, A. Abraham, M. Chis, Swarm intelligence in data mining Studies in Computational Intelligence, vol. 34, Springer, 2006, pp. 1–20. Q3
[7] D. Martens, B. Baesens, T. Fawcett, Editorial survey: swarm intelligence for data mining, Machine Learning 82 (2011) 1–42.
[8] Z.X. Huang, Extensions to the k-means algorithms for clustering large data sets with categorical values, Data mining and Knowledge Discovery 2 (1998) 283–304.
[9] T. Kanungo, D.M. Mount, N.S. Netanyahu, An efficient k-means clustering algorithm: analysis and implementation, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 881–892.
[10] T. Zhang, L.M. Ramakrishna, BRICH: an efficient data clustering method for very large databases, in: ACM SIGMOD International Conference on Management of Data, ACM Press, New York, 1996, pp. 103–114.
[11] R.M. Aliguliyev, Performance evaluation of density-based clustering methods, Information Sciences 179 (2009) 3583–3602.
[12] L. Duan, L. Xu, F. Guo, A local-density based spatial clustering algorithm with noise, Information Systems 32 (2007) 978–986.
[13] S.H. Yue, M.M. Wei, J.S. Wang, A general grid-clustering approach, Pattern Recognition Letters 29 (9) (2008) 1372–1384.
[14] D. Brugger, M. Bogdan, W. Rosenstiel, Automatic clustering detection in Kohonen's SOM, IEEE Transactions on Neural Networks 19 (3) (2008) 442–459.
[15] S.K. Rangarajan, V.V. Phoha, K.S. Balagani, Adaptive neural network clustering of Web users, Computer 37 (4) (2004) 34–40.
[16] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Analytica Chimica Acta 509 (2004) 187–195.
[17] L. Chen, L. Tu, H.J. Chen, A novel ant clustering algorithm with digraph Lecture Notes in Computer Science, vol. 3611, Springer, Berlin, Heidelberg, 2005, pp. 1218–1228.
[18] S. Ghosh, M. Kothari, A. Halder, A. Ghosh, Use of aggregation pheromone density for image segmentation, Pattern Recognition Letters 30 (2009) 939–949.
[19] D. Marco, B. Eric, T. Guy, Ant algorithms and stigmergy, Future Generation Computer System 16 (2000) 851–871.
[20] J.L. Deneubourg, S. Goss, N. Frank, The dynamics of collective sorting: robot-like ants and ant-like robots, in: Proceedings of 1st International Conference on Simulation of Adaptive Behavior: From Animals to Animats, MIT Press/Bradford Books, Cambridge, MA, 1991, pp. 356–363.

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022

[21] E. Lumer, B. Faieta, Diversity and adaptation in populations of clustering ants, in: 3rd Interantional Conference on Simulation of Adaptive Behavior: From Animals to Animats, MIT Press/Bradford Books, Cambridge, MA, 1994, pp. 501–508.

[22] B. Wu, Y. Zheng, S. Liu, Z.Z. Shi, CSIW: a document clustering algorithm based on swarm intelligence, vol. 1, in: Proceeding of the 2002 Congress on Evolutionary Computation, Honolulu, HI, USA, 2002, pp. 477–482.

[23] V. Ramos, J.J. Merelo, Self-organized stigmergic document maps: environment as a mechanism for context learning, in: 1st Spanish Conference on Evolutionary and Bio-Inspired Algorithms, Centro Uni. De Merida, Spain, 2002, pp. 284–293.

[24] Y. Yang, M. Kamel, F. Jin, Topic discovery from document using ant-based clustering combination Lecture Notes in Computer Science, vol. 3399, Springer Berlin, Heidelberg, 2005, pp. 100–108.

[25] Y. Yang, M. Kamel, An aggregated clustering approach using multi-ant colonies algorithms, Pattern Recognition 39 (2006) 1278–1289.

[26] J. Handl, B. Meyer, Improved ant-based clustering and sorting in document retrieval interface Lecture Notes in Computer Science, vol. 2439, Springer Berlin, Heidelberg, 2002, pp. 913–923.

[27] J. Handl, J. Knowles, M. Dorigo, Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and 1D-SOM. Technical Report TR/IRIDIA/2003-24, IRIDIA, University Libre de Bruxelles, Belgium, 2003.

[28] J. Handl, J. Knowles, M. Dorigo, Strategies for the increased robustness of ant-based clustering, Lecture Notes in Artificial Intelligence, vol. 2977, Springer-Verlag, Berlin Heidelberg, 2004, pp. 90–104.

[29] J. Handl, J. Knowles, M. Dorigo, Ant-based clustering and topographic mapping, Artificial Life 12 (1) (2004) 1–36.

[30] J. Handl, B. Meyer, Ant-based and swarm-based clustering, Swarm Intelligence 1 (2) (2007) 95–113.

[31] C.H. Tsang, S. Kwong, Ant colony clustering and feature extraction for anomaly intrusion detection Studies in Computing Intelligence, vol. 3, Springer Berlin, Heidelberg, 2006, pp. 101–123.

[32] X.H. Xu, L. Chen, Y.X. Chen, A4C: an adaptive artificial ants clustering algorithm, in: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2004, pp. 268–274.

[33] N. Monmarche, M. Slimane, G. Venturini, AntClass: discovery of clusters in numeric data by a hybridization of an ant colony with the k-means algorithm. Internal Report No. 213 [Online] Available from: http://www.antsearch.univ-tours.fr/public/MonSliVen99b.pdf (1999).

[34] H. Jiang, S. Yi, J. Li, F. Yang, X. Hu, Ant clustering algorithm with K-harmonic means clustering, Expert Systems with Applications 37 (12) (2010) 8679–8684.

[35] Z. Yu, O.C. Au, R. Zou, W. Yu, J. Tian, An adaptive unsupervised approach toward pixel clustering and color image segmentation, Pattern Recognition 43 (2010) 1889–1906.

[36] E. Lefever, T. Fayruzov, V. Hoste, M. De Cock, Clustering web people search results using fuzzy ants, Information Sciences 180 (2010) 3192–3209.

[37] T. Niknam, B. Amiri, An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis, Applied Soft Computing 10 (2010) 183–197.

[38] B. Liu, J.H. Pan, R.I. McKay, Entropy-based metrics in swarm clustering, International Journal of Intelligent Systems 24 (2009) 989–1011.

[39] U. Boryczka, Finding groups in data: cluster analysis with ants, Applied Soft Computing 9 (2009) 61–70.

[40] G. Erhan, J.C. Principe, Information theoretic clustering, IEEE Transactions on Pattern Analysis 24 (2) (2002) 158–171.

[41] R. Jenssen, K.E. Hild I.I., D. Erdogmus, et al., Clustering using Renyi's entropy, International Joint Conference on Neural Networks 1 (2003) 523–528.

[42] A. Renyi, On measure of entropy and information, in Fourth Berkeley Symposium on Mathematical Statistics and Probability (1960) 547–561.

[43] J.C. Principe, D.X. Xu, J.W. Fisher, Information theoretic learning, in: S. Haykin (Ed.), Unsupervised Adaptive Filtering, John Wiley &Sons, New York, 2000, pp. 1–62.

[44] G. Erhan, J.C. Principe, A new clustering evaluation function using Renyi's information potential, in: International Conference on Acoustics, Speech, and Signal Processing, 2000, pp. 3491–3493.

[45] I.T. Jolliffe, Principal Component Analysis, Springer Verlag, 1986.

[46] B. Scholkopf, A. Smola, K.R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computing 10 (5) (1998) 1299–1319.

[47] R. Jenssen, Kernel entropy component analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (5) (2010) 847–860.

[48] L. Zhang, Q.X. Cao, J. Lee, A modified clustering algorithm based on swarm intelligence, in: The First International Conference on Natural Computation, vol. 3, Changsha, China, 2005, pp. 535–542.

[49] L. Zhang, Q.X. Cao, A novel ant-based clustering algorithm using the kernel method, Information Sciences 118 (20) (2011) 4658–4672.

[50] M. Girolami, Mercer kernel-based clustering in feature space, IEEE Transactions on Neural Networks 13 (2) (2002) 780–784.

[51] R. Jenssen, T. Eltoft, A new information theoretic analysis of sum-of-squared-error kernel clustering, Neurocomputing 72 (2008) 23–31.

[52] <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

[53] <http://dbkgroup.org/handl/ants/>.

[54] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognition 41 (1) (2008) 176–190.

[55] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[56] R.L. Iman, J.M. Davenport, Approximations of the critical region of the Friedman statistic, Communications in Statistics (1980) 571–595.

Please cite this article in press as: L. Zhang, et al., A novel ant-based clustering algorithm using Renyi entropy, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.11.022